



**Rui Pedro
Anastácio Barbosa**

**Reconhecimento de gestos para apresentações
informáticas interativas**



**Rui Pedro
Anastácio Barbosa**

**Reconhecimento de gestos para apresentações
informáticas interativas**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestrado em Engenharia de Automação Industrial, realizada sob a orientação científica do Engenheiro Abílio Manuel Ribeiro Borges, Assistente Convidado do Departamento de Engenharia Mecânica da Universidade de Aveiro.

o júri

presidente

Prof. Doutor Pedro Nicolau Faria da Fonseca

Professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

arguente

Doutor Miguel Armando Riem de Oliveira

Bolseiro de Pós-Doutoramento do Instituto de Engenharia Eletrónica e Telemática de Aveiro

orientador

Engenheiro Abílio Manuel Ribeiro Borges

Assistente Convidado do Departamento de Engenharia Mecânica da Universidade de Aveiro

**agradecimentos /
acknowledgements**

Agradeço ao Engenheiro Abílio Borges pela orientação dada durante o desenvolvimento desta dissertação. Pela ajuda e motivação disponibilizada ao longo do tempo, sem a qual seria possível realizar.

Um agradecimento a toda a minha família que sempre esteve presente, em especial à minha mãe pelo esforço, dedicação, confiança e proporcionar todas as condições necessárias para que fosse possível tornar-me na pessoa que sou hoje.

Agradeço, também a todos os meus amigos e colegas com o qual tive o privilégio de conviver e trabalhar, pois permitiram que este caminho de esforço e dedicação fosse também de alegria, bons momentos e acima de tudo amizade. Um muito obrigado a todas estas pessoas, pois sem eles não teria sido possível ter chegado até aqui.

palavras-chave

Quadro interativo, Kinect, Reconhecimento de Gestos, Apresentações interativas, Reconhecimento pele humana, C#, Interface Homem-Máquina, Comandos voz.

resumo

Este documento apresenta e detalha um interface desenvolvido que permite apresentações em quadros interativos através de gestos realizados com as mãos recorrendo ao sensor Kinect.

Com o aumento da procura de interfaces que permitam uma melhor interatividade entre o homem e este tipo de tecnologia, este trabalho tenta abordar uma solução diferente às existentes no mercado através do reconhecimento da pele humana, uma característica com muita importância nos dias de hoje.

O interface proposto permite a um utilizador atuar num computador através de gestos produzidos com as suas mãos e com isso realizar os mais diferentes tipos de ações. O controlo por voz, desenho e escrita foram também particularidades estudadas e desenvolvidas, pois são funcionalidades que permitem melhorar a interatividade com o utilizador.

keywords

Interactive whiteboard, Kinect, Gesture recognition, Interactive presentations, Human skin recognition, C#, Human-machine interface, Voice command.

abstract

This document provides and detail a developed interface that enables presentations on interactive whiteboards through gestures made with hands, using the Kinect sensor.

With the increasing demand for interfaces that enable better interaction between man and this type of technology, this thesis attempts to address a different solution to the existing market by recognizing human skin, a characteristic very importance these days. This interface allows a user to operate a computer with gestures produced with hands and perform different types of actions. The voice control, drawing and writing were also particularities studied and developed, they are features that improve the interactivity with the user.

Conteúdos

Índice de Figuras	iv
Índice de Tabelas	ix
Índice de Equações	xi
Capítulo 1	1
Introdução	1
1.1 Enquadramento	1
1.2 Motivação e Objetivos	3
1.3 Estrutura da Dissertação	4
Capítulo 2	5
Estado da Arte	5
2.1 Introdução	5
2.2 Tecnologias utilizadas	6
2.2.1 Resistiva	6
2.2.2 Eletromagnética	7
2.2.3 Capacitiva	7
2.2.4 Leitura ótica de infravermelhos	8
2.2.5 Ultrassons	9
2.2.6 DVIT	9
2.3 Métodos de projeção	10
2.3.1 Projeção frontal	10
2.3.2 Projeção traseira	11
2.3.3 Ecrã LCD ou Plasma	11
2.4 Áreas de utilização	12
2.4.1 Quadros interativos nas escolas	12
2.4.2 Quadros interativos nas empresas	14
Capítulo 3	15
Arquitetura	15
3.1 Microsoft Kinect	15
3.1.1 O sensor Kinect	15
3.1.2 SDK (Software Development Kit)	17
3.1.3 Video Stream	18
3.1.4 Depth Stream	19

3.1.5	Rastreamento do esqueleto	21
3.2	Emgu CV	22
3.3	Arquitetura do Sistema	23
3.4	Projetos desenvolvidos.....	26
3.4.1	Jintronix.....	26
3.4.2	Skanect	27
3.4.3	Ikkos.....	28
3.4.4	Kimetric	29
3.4.5	Picksurelt.....	29
3.4.6	Styku	30
3.4.7	Ubi Interactive	30
3.4.8	Atlas5D.....	31
3.4.9	GestSure Technologies	32
3.4.10	Kinect Hand Detection.....	32
Capítulo 4		34
Desenvolvimento.....		34
4.1	Espaços de cor para o reconhecimento da pele.....	34
4.1.1	Espaço RGB	34
4.1.2	Espaço HSV	35
4.1.3	Espaço YCbCr	37
4.2	Reconhecimento e deteção das mãos	38
4.2.1	Introdução	38
4.2.2	Segmentação pela cor da pele.....	39
4.2.2.1	Estudo HSV	40
4.2.2.2	Estudo YCbCr	45
4.2.2.3	Método probabilístico	49
4.2.3	Segmentação pela profundidade	55
4.2.4	Deteção das mãos e extração das suas caraterísticas	57
4.2.4.1	Área da mão.....	57
4.2.4.2	Centro da mão.....	58
4.2.4.3	Deteção dos dedos	58
4.2.4.4	Deteção das duas mãos.....	61
4.3	Classificador de gestos e movimentos	62
4.4	Atuação de gestos.....	65

Capítulo 5	67
Aplicação e análise de resultados	67
5.1 Introdução	67
5.2 Aquisição de dados	68
5.3 Correção do offset	69
5.4 Segmentação pela cor da pele	72
5.5 Extração das características da mão	76
5.6 Detecção das duas mãos	78
5.7 Reconhecimento de gestos	79
5.8 Atuação	83
5.9 Desenho e escrita interativa	85
5.10 Comando por voz	86
Capítulo 6	88
Conclusões e trabalho futuro	88
6.1 Conclusão	88
6.2 Trabalho Futuro	89
Referências Bibliográficas	91
Apêndice 1	95
Ensaios HSV, YUV e Método Probabilístico	95

Índice de Figuras

Figura 1 – Infografia publicada pela empresa GO-Globe.com sobre o consumo de produtos.....	2
Figura 2 – Ecrã interativo baseado na tecnologia resistiva.....	6
Figura 3 – Ecrã interativo baseado na tecnologia capacitiva.....	8
Figura 4 – Ecrã interativo baseado na tecnologia infravermelhos.....	9
Figura 5 – Quadros interativo de projeção frontal de curta distância.....	10
Figura 6 – Quadro interativo de projeção traseira.....	11
Figura 7 – Quadro interativo utilizando um ecrã.....	11
Figura 8 – Percentagem de professores que utiliza os recursos digitais de aprendizagem, pelo menos em metade das aulas.....	13
Figura 9 – Identificação de componentes externos do sensor Kinect.....	15
Figura 10 – Limites do sensor Kinect.....	16
Figura 11 – Arquitetura do SDK Kinect.....	17
Figura 12 – Pontos capturados pela câmara profundidade.....	19
Figura 13 – Imagem profundidade adquirida pela Kinect.....	20
Figura 14 – Bits de um píxel do stream profundidade.....	20
Figura 15 – Conjunto de juntas detetadas pelo sensor Kinect.....	21
Figura 16 – Sistema de coordenadas das juntas do esqueleto.....	22
Figura 17 – Diagrama de blocos do sistema a desenvolver.....	25
Figura 18 – Sistema de reabilitação Jintronix.....	27
Figura 19 – Exemplo modelo 3D Skanect.....	28
Figura 20 – Equipamento Ikkos.....	28
Figura 21 – Software Picksurelt.....	29
Figura 22 – Software Atlas5D.....	31
Figura 23 – Software GestSure.....	32
Figura 24 – Kinect Hand Detection Software.....	33
Figura 25 – Sistema de cores aditivo e subtrativo.....	35
Figura 26 – Cubo RGB.....	35
Figura 27 – Modelo hexagonal representativo do espaço HSV.....	36
Figura 28 – Imagem RGB.....	40
Figura 29 – Imagem HSV.....	40
Figura 30 – Programa Matlab para análise do espaço de cor HSV.....	41
Figura 31 – Histograma HSV.....	41
Figura 32 – Imagem da segmentação efetuada pelo espaço de cor HSV.....	42
Figura 33 – Imagem final obtida após a segmentação HSV.....	42
Figura 34 – Gráfico do parâmetro tonalidade.....	44
Figura 35 – Gráfico do parâmetro saturação.....	44
Figura 36 – Gráfico do parâmetro brilho.....	45
Figura 37 – Imagem RGB.....	46
Figura 38 – Imagem YCbCr.....	46
Figura 39 – Histograma YCbCr.....	46
Figura 40 – Imagem da segmentação efetuada pelo espaço de cor YCbCr.....	47
Figura 41 – Imagem final obtida após a segmentação YCbCr.....	47
Figura 42 – Gráfico do parâmetro cromaticidade Cb.....	48
Figura 43 – Gráfico do parâmetro cromaticidade Cr.....	48
Figura 44 – Representação da gama de cores dos píxeis skin.....	49
Figura 45 – Representação de píxeis skin e nonskin.....	49
Figura 46 – Representação da probabilidade da componente R ser ou não skin.....	50
Figura 47 – Representação da função de densidade de probabilidade.....	50

Figura 48 – Teorema de Bayes aplicado ao nosso problema.....	51
Figura 49 – Curvas ROC (Receiver Operating Characteristic) para diferentes representações de cor. (a) Histograma 256. (b) Histograma 64.....	52
Figura 50 – Imagem RGB.....	53
Figura 51 – Imagem método Bayes.....	54
Figura 52 – Imagem segmentada pelo método Bayes.....	54
Figura 53 – Representação da zona de atuação na vista frontal.....	55
Figura 54 – Representação da zona de atuação na vista lateral.....	56
Figura 55 – Imagem profundidade.....	56
Figura 56 – Imagem da segmentação por profundidade.....	56
Figura 57 – Representação da caixa e localização da palma da mão.....	58
Figura 58 – Representação dos pontos característicos.....	58
Figura 59 – Representação do cálculo dos Convexity defects.....	59
Figura 60 – Representação do conjunto de pontos de interesse e não interesse.....	60
Figura 61 – Método para deteção dos pontos correspondentes às pontas dos dedos.....	60
Figura 62 – Inicialização dos eventos para receção das imagens.....	68
Figura 63 – Eventos para receção das imagens.....	68
Figura 64 – Aquisição da imagem RGB.....	68
Figura 65 – Aquisição da imagem Profundidade.....	69
Figura 66 – Offset vertical entre a imagem RGB e infravermelhos.....	70
Figura 67 – Offset horizontal entre a imagem RGB e infravermelhos.....	70
Figura 68 – Modo de calibração a calcular erro.....	71
Figura 69 – Modo de calibração finalizado.....	72
Figura 70 – Fluxograma do processo de segmentação pela cor da pele.....	72
Figura 71 – Exemplo de matriz multidimensional utilizada.....	73
Figura 72 – Imagem RGB sensor Kinect.....	73
Figura 73 – Imagem do resultado da segmentação pela cor da pele.....	74
Figura 74 – Imagem binarizada após tratamento.....	75
Figura 75 – Imagem segmentação por pele.....	75
Figura 76 – Imagem antes da operação lógica.....	76
Figura 77 – Imagem final após operação lógica.....	76
Figura 78 – Área localizada da mão.....	76
Figura 79 – Contorno da mão.....	77
Figura 80 – Centro e contorno convexo da mão.....	77
Figura 81 – Identificação dos dedos da mão.....	78
Figura 82 – Identificação de duas mãos em simultâneo.....	78
Figura 83 – Classe <i>Entry</i>	79
Figura 84 – Classe <i>SwipeGestureDetector</i>	80
Figura 85 – Método <i>LookForGesture</i>	80
Figura 86 – Ponto inicial movimento de avanço.....	81
Figura 87 – Ponto final movimento de avanço.....	81
Figura 88 – Esquema de validação gesto de avanço.....	82
Figura 89 – Inicialização do ficheiro Powerpoint.....	83
Figura 90 – Inicialização <i>VoiceCommaner</i>	86
Figura 91 – Inicialização da captura do som.....	86
Figura 92 – Operações para as palavras detetadas.....	87
Figura 93 – Transformação RGB para HSV da mão número 1.....	95
Figura 94 – Histograma HSV da imagem da mão número 1.....	95
Figura 95 – Segmentação HSV da mão número 1.....	95
Figura 96 – Transformação RGB para YCbCr da mão número 1.....	96
Figura 97 – Histograma YCbCr da mão número 1.....	96
Figura 98 – Segmentação YCbCr da mão número 1.....	96

Figura 99 – Transformação de RGB para método probabilístico da mão número 1.	97
Figura 100 – Resultado das operações morfológicas na imagem binarizada da mão número 1.	97
Figura 101 – Imagem da mão número 1 obtida após método probabilístico.	97
Figura 102 – Transformação RGB para HSV da mão número 2.	98
Figura 103 – Histograma HSV da mão número 2.	98
Figura 104 – Segmentação HSV da mão número 2.	98
Figura 105 – Transformação RGB para YCbCr da mão número 2.	99
Figura 106 – Histograma YCbCr da mão número 2.	99
Figura 107 – Segmentação YCbCr da mão número 2.	99
Figura 108 – Transformação de RGB para método probabilístico da mão número 2.	100
Figura 109 – Resultado das operações morfológicas na imagem binarizada da mão número 2.	100
Figura 110 – Imagem da mão número 2 obtida após método probabilístico.	100
Figura 111 – Transformação RGB para HSV da mão número 3.	101
Figura 112 – Histograma HSV da mão número 3.	101
Figura 113 – Segmentação HSV da mão número 3.	101
Figura 114 – Transformação RGB para YCbCr da mão número 3.	102
Figura 115 – Histograma YCbCr da mão número 3.	102
Figura 116 – Segmentação YCbCr da mão número 3.	102
Figura 117 – Transformação de RGB para método probabilístico da mão número 3.	103
Figura 118 – Resultado das operações morfológicas na imagem binarizada da mão número 3.	103
Figura 119 – Imagem da mão número 3 obtida após método probabilístico.	103
Figura 120 – Transformação RGB para HSV da mão número 4.	104
Figura 121 – Histograma HSV da mão número 4.	104
Figura 122 – Segmentação HSV da mão número 4.	104
Figura 123 – Transformação RGB para YCbCr da mão número 4.	105
Figura 124 – Histograma YCbCr da mão número 4.	105
Figura 125 – Segmentação YCbCr da mão número 4.	105
Figura 126 – Transformação de RGB para método probabilístico da mão número 4.	106
Figura 127 – Resultado das operações morfológicas na imagem binarizada da mão número 4.	106
Figura 128 – Imagem da mão número 4 obtida após método probabilístico.	106
Figura 129 – Transformação RGB para HSV da mão número 5.	107
Figura 130 – Histograma HSV da mão número 5.	107
Figura 131 – Segmentação HSV da mão número 5.	107
Figura 132 – Transformação RGB para YCbCr da mão número 5.	108
Figura 133 – Histograma YCbCr da mão número 5.	108
Figura 134 – Segmentação YCbCr da mão número 5.	108
Figura 135 – Transformação de RGB para método probabilístico da mão número 5.	109
Figura 136 – Resultado das operações morfológicas na imagem binarizada da mão número 5.	109
Figura 137 – Imagem da mão número 5 obtida após método probabilístico.	109
Figura 138 – Transformação RGB para HSV da mão número 6.	110
Figura 139 – Histograma HSV da mão número 6.	110
Figura 140 – Segmentação HSV da mão número 6.	110
Figura 141 – Transformação RGB para HSV da mão número 6.	111
Figura 142 – Histograma YCbCr da mão número 6.	111
Figura 143 – Segmentação YCbCr da mão número 6.	111
Figura 144 – Transformação de RGB para método probabilístico da mão número 6.	112

Figura 145 – Resultado das operações morfológicas na imagem binarizada da mão número 6.	112
---	-----

Índice de Tabelas

Tabela 1 – Resultados dos intervalos de valores HSV.....	42
Tabela 2 – Resultados dos intervalos de valores YCbCr.	47
Tabela 3 – Taxas de classificação (CRs) de oito espaços de cor (Tamanho Histograma = 256).	52
Tabela 4 – Área da mão relativamente à distância em que se encontra.	57
Tabela 5 – Comprimento dos dedos da mão.	61
Tabela 6 – Lista de gestos e atuações.....	65
Tabela 7 – Descrição de tipos de comandos por voz.	66
Tabela 8 – Operações morfológicas do Emgu CV.	74
Tabela 9 – Percorso de posições sucessivas do gesto de avanço.....	82
Tabela 10 – Código de atuação de gestos no sistema.....	84

Índice de Equações

Equação 1 – Cálculo da tonalidade através dos parâmetros RGB.....	37
Equação 2 – Cálculo da saturação através dos parâmetros RGB.....	37
Equação 3 – Cálculo do brilho através dos parâmetros RGB.....	37
Equação 4 – Cálculo da luminância através dos parâmetros RGB.	38
Equação 5 – Cálculo da cromaticidade C_b através dos parâmetros RGB.....	38
Equação 6 – Cálculo da cromaticidade C_r através dos parâmetros RGB.	38
Equação 7 – Probabilidade de um píxel ser nonskin.....	50
Equação 8 – Teorema de Bayes.....	51
Equação 9 – Logaritmo da probabilidade de um píxel ser skin.	53

Capítulo 1

Introdução

1.1 Enquadramento

A evolução tecnológica sempre esteve presente na vida do homem e assume uma elevada importância quer para melhorar quer para simplificar o seu modo de vida.

Nos nossos dias existe um aumento crescente na sua relação com a tecnologia e no seu ritmo de evolução. Vive-se numa época em que tudo é muito rápido e é necessário responder e apresentar soluções que satisfaçam as suas necessidades.

A principal razão pela qual se investe na tecnologia é a permanente procura do aumento da produtividade, eficiência e acima de tudo num maior retorno financeiro. É impossível não associar cada segundo que o homem leva a executar determinada tarefa, a um fator financeiro ou ao tempo que ocupa no seu dia-a-dia.

Nos últimos anos, a tecnologia tem evoluído tanto que é quase impensável que em qualquer país desenvolvido exista alguém que direta ou indiretamente, não tenha entrado em contacto com sistemas computacionais. O leque de utilizadores tem aumentado e já não se restringe apenas a pessoas na área da Informática ou Eletrónica, incluindo pessoas das mais variadas áreas de conhecimento.

A Go-Globe.Com, empresa que trabalha com *web design* internacional, elaborou uma infografia [1] que permite visualizar o consumo de diversos produtos digitais ou relacionados à internet a cada 60 segundos como se verifica na Figura 1.



Figura 1 – Infografia publicada pela empresa GO-Globe.com sobre o consumo de produtos.

Adaptada de [1].

Um fator importante na área da tecnologia é o modo em que o ser humano se interliga com ela. Desde cedo surgiu essa preocupação por parte da indústria, de modo a que ela seja projetada por forma a atender às necessidades e se adaptar às capacidades das pessoas a quem ela é dirigida. A cada dia que passa, o homem torna-se mais dependente da tecnologia, o que provoca um aumento da sua exigência com a qualidade da mesma, tornando-se menos tolerável à perda de tempo na utilização de recursos tecnológicos.

O Homem interage com o computador utilizando os periféricos tão bem conhecidos como o rato e teclado, fundamentais nos tempos de hoje, mas que começam a ser vistos como uma tecnologia do passado. Os quadros interativos são uma tecnologia atual e tendem a renovar o método como se realiza essa interação.

Existem diversos métodos de comunicação com os quadros interativos e a tendência é sempre de evoluir para que no futuro seja possível comunicar com um computador sem recorrer a qualquer dispositivo, sendo possível realizar uma operação, através de reconhecimento de gestos e voz

1.2 Motivação e Objetivos

A motivação para realizar este trabalho surge do interesse do autor pela tecnologia, multimédia e visão computacional. A possibilidade de desenvolver um interface inteligente que permita ao ser humano uma interação mais natural e intuitiva com a tecnologia, constituiu um fator determinante para a escolha do tema da dissertação.

A importância que a tecnologia de quadros interativos ostenta no mercado atual tem vindo a apresentar um aumento crescente nos últimos anos nas mais diversas áreas como a educação ou meio empresarial. São diversas as soluções disponíveis no mercado, mesmo assim, estas apresentam um valor demasiado elevado para a maioria dos consumidores que gostariam de usufruir deste tipo de tecnologia.

O objetivo principal deste trabalho é desenvolver um método inovador de comunicação entre o quadro interativo e o utilizador, recorrendo ao uso do sensor Kinect. O Kinect tem vindo a ser utilizado em diversas aplicações e já existem diversas soluções que o tornam numa ferramenta extraordinária na transformação de superfícies em superfícies interativas.

Sendo o reconhecimento da pele humana uns dos temas mais investigados nos tempos atuais, a solução apresentada neste trabalho permitiu explorar uma alternativa inovadora, até aqui pouco conseguida na deteção das mãos do utilizador através da pele humana, de uma forma mais abrangente possível. Também foi implementado o reconhecimento de um grupo de gestos que permitissem ao utilizador uma forma de atuação num dispositivo interativo.

Uma das principais vantagens que esta solução apresenta no mercado é diminuir as restrições a nível de espaço de atuação, pois o controlo não é feito a partir da deteção do esqueleto como a maioria dos interfaces disponíveis, mas sim da identificação da cor da pele.

O desenvolvimento de aplicativos de baixo custo que permitam a deteção da pele, reconhecendo e interpretando os movimentos de um humano de forma a atuar num dispositivo eletrónico, torna-se então numa realidade mais próxima de se concretizar.

1.3 Estrutura da Dissertação

Este documento está dividido por 6 capítulos principais.

No capítulo 1, é realizada uma introdução com um enquadramento do tema que será estudado. É apresentada a motivação e os objetivos para a realização da Tese de Mestrado.

No capítulo 2, apresenta-se um estudo efetuado sobre as diversas tecnologias de quadros interativos existentes na atualidade e a sua importância em diversas áreas de utilização.

No capítulo 3, é descrita a arquitetura utilizada para se proceder ao desenvolvimento do sistema, contém uma introdução ao sensor Kinect e são explicadas detalhadamente todas as suas funcionalidades. É feita uma breve introdução à biblioteca utilizada para extrair as características das mãos, e é ainda apresentado um diagrama da arquitetura do sistema. Por fim são descritos alguns exemplos de interfaces existentes no mercado que recorrem à utilização da arquitetura Kinect.

No capítulo 4, são expostos os problemas que surgiram no desenvolvimento do sistema e explicada a sua resolução. Inicia-se com o estudo dos espaços de cor existentes e a metodologia adotada para realizar a deteção da cor da pele. De seguida é demonstrado todo o processo de reconhecimento, deteção e extração das características das mãos. Por fim é apresentado o método utilizado para classificar os gestos e a sua atuação no sistema.

No capítulo 5, é realizada uma análise aos resultados obtidos em todos os processos do desenvolvimento do sistema.

Por fim, no capítulo 6, é descrita a conclusão do trabalho e de uma forma geral, os resultados obtidos. Analisar-se-á se os objetivos inicialmente propostos foram atingidos e que melhoramentos poderão vir a implementar no futuro.

Capítulo 2

Estado da Arte

2.1 Introdução

O primeiro quadro interativo apresentado ao mundo foi produzido pela empresa *SMART Technologies Inc.* em 1991. Esta empresa foi fundada em 1987 por David Martin e Nancy Knowlton, atualmente presidente e vice-presidente respetivamente.

As potencialidades desta tecnologia foi rapidamente reconhecida por muitos como uma tecnologia de futuro a qual iria mudar radicalmente o modo como o utilizador iria interagir com o computador.

Esta tecnologia é usualmente utilizada em apresentações interativas não só em salas de aulas, mas também no meio empresarial na realização de reuniões e no desporto para a apresentação de jogadas, representação de táticas bem como outras aplicações.

O sistema quadro interativo engloba três componentes: um computador responsável por gerar a imagem, um projetor digital que realiza a sua projeção e o quadro interativo que consiste numa superfície digital no qual é projetada a imagem.

Na superfície do quadro é projetado o ambiente de trabalho do computador e dependendo da tecnologia utilizada, é possível realizar o controlo do computador de diversas formas, nomeadamente, eletromagnética, sensível ao toque ou infravermelhos.

O uso deste dispositivo permite a interação do utilizador com qualquer que seja a aplicação utilizada, desde que esta funcione corretamente no computador, sendo que existem aplicações específicas para este tipo de equipamento como por exemplo *software* didático para dar aulas, reconhecer escrita e de o converter para formato digital.

A ligação entre o quadro interativo e o computador pode ser feito por cablagem via USB, cabo de porta série ou ligação sem fios *Bluetooth*.

Após a inicialização do computador o quadro interativo encontra-se disponível para o seu correto funcionamento. Existe um driver, acoplado ao computador, responsável por realizar a conversão do contacto ou posicionamento em relação ao quadro interativo em operações do ponteiro do rato ou tinta digital.

2.2 Tecnologias utilizadas

Como foi referido anteriormente, os quadros interativos podem ser baseados em várias tecnologias. As mais comuns e relevantes são [2]:

2.2.1 Resistiva

Nesta tecnologia são utilizadas duas superfícies de material condutor que se separam entre si por uma camada de ar. Quando pressionada a superfície exterior, as duas camadas entram em contacto deixando de existir a lacuna de ar que as separa originando uma mudança de resistência entre essas duas camadas. Este contacto é convertido num dado digital que é enviado para o computador e processado, o qual permite identificar as coordenadas XY em que esse ponto de contacto foi estabelecido. A Figura 2 ilustra a tecnologia descrita.

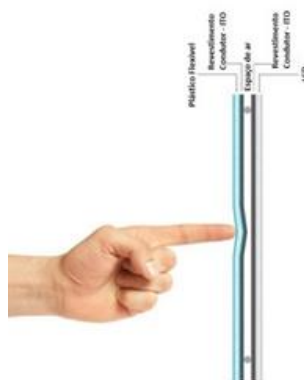


Figura 2 – Ecrã interativo baseado na tecnologia resistiva. Adaptada de [3].

A vantagem deste tipo de tecnologia é o controlo poder ser efetuado com um simples dedo, caneta ou dispositivo próprio para o efeito. No entanto também apresenta as suas desvantagens nomeadamente o facto de ser apenas possível um ponto de contacto de cada vez, ou seja, não existe possibilidade de operações *multi-touch*, a sensibilidade e precisão é relativamente inferior comparando a outros tipos de tecnologias e com o passar do tempo apresenta um maior desgaste relativamente à sua precisão, o que diminui a sua durabilidade.

Esta tecnologia torna-se um pouco difícil de ser utilizada por pessoas muito jovens ou pouco experientes porque qualquer tipo de contacto com a superfície aquando da operação

do mesmo, seja com a mão ou qualquer outro tipo de dispositivo vai afetar o trabalho que a pessoa está a realizar.

2.2.2 Eletromagnética

Na tecnologia eletromagnética os quadros são mais robustos, a sua superfície de contacto e estrutura são fabricados com materiais mais rígidos.

O modo como esta tecnologia funciona é a seguinte, um conjunto de fios embutidos atrás da superfície de contacto interage com uma bobine presente num dispositivo próprio, normalmente uma caneta. Através deste sistema é possível adquirir as coordenadas X e Y da mesma em relação ao quadro.

A caneta para efetuar o controlo pode ser de dois tipos: ativa ou passiva. A diferença principal está que a tecnologia ativa necessita de alimentação do dispositivo, seja por bateria, pilhas, ou um cabo proveniente do quadro, a tecnologia passiva não requer qualquer tipo de alimentação tendo esta tecnologia a mesma capacidade de alterar os sinais elétricos produzidos pelo quadro.

Este contacto entre a caneta e o quadro produz sinais elétricos que vão ser lidos através dos sensores eletromagnéticos embutidos no quadro, responsáveis por enviar essa informação para o computador.

Este tipo de tecnologia tem a vantagem de permitir que o utilizador entre em contacto com o quadro através da mão quando escreve, com a caneta ou com o dedo para apontar, sem que isso interfira no seu funcionamento.

À semelhança das salas de aulas, esta tecnologia é muito utilizada na área de *design* ou desenhadores profissionais e normalmente suportam resoluções mais elevadas.

2.2.3 Capacitiva

Os quadros capacitivos apresentam uma tecnologia semelhante à anterior, e consiste num barramento de fios em duas placas com uma orientação de 90 graus entre si como é representado na Figura 3.

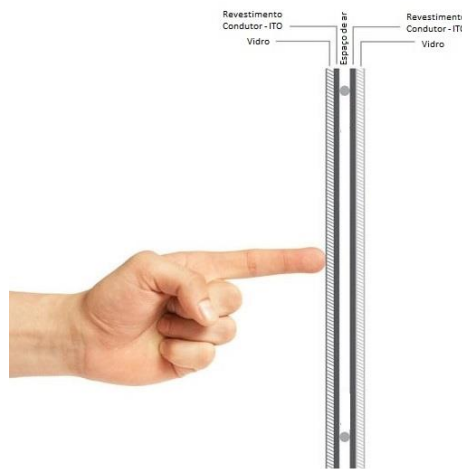


Figura 3 – Ecrã interativo baseado na tecnologia capacitiva. Adaptada de [3].

A pressão num ponto do ecrã pode ser efetuada por um dedo ou outro tipo de dispositivo próprio, o que provoca o contacto entre dois fios, um na orientação do eixo X e outro Y. Esse contacto altera o campo eletrostático o que permite processar essa informação e calcular a posição onde foi exercido o toque.

Esta tecnologia é mais precisa e apresenta as vantagens de não ser necessário recorrer a nenhum dispositivo para interagir com o quadro, permitir reconhecimento *multi-touch* e a sua durabilidade ser mais elevada.

2.2.4 Leitura ótica de infravermelhos

Na tecnologia por infravermelhos é efetuada a leitura da posição da caneta ou do dedo na superfície do quadro interativo a partir da interrupção que ocorre nos feixes infravermelhos ao longo de todo o quadro na orientação horizontal e vertical.

O quadro pode ser construído em qualquer tipo de material, visto que este tipo de leitura é efetuada independentemente do material em que é fabricado.

A Figura 4 ilustra a tecnologia de infravermelhos.

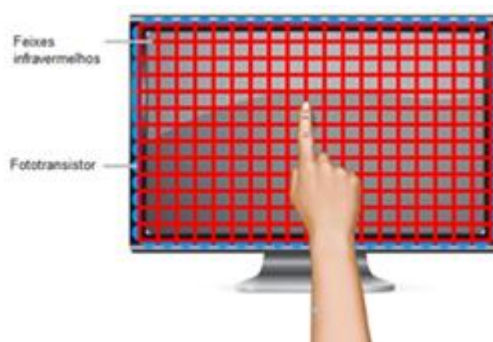


Figura 4 – Ecrã interativo baseado na tecnologia infravermelhos. Adaptada de [4].

2.2.5 Ultrassons

Quando uma caneta própria para este tipo de tecnologia entra em contacto com o quadro, esta envia um som ultrassónico que irá ser rececionado por microfones instalados em posições pré-definidas. O cálculo da posição da caneta é efetuado a partir do tempo de receção de cada sinal recetor e a diferença entre si. Esta tecnologia não impõe que o quadro seja fabricado por qualquer tipo de material específico, mas apresenta a desvantagem de apenas funcionar com uma caneta própria e impossibilitar o uso dos dedos para efetuar o seu controlo.

2.2.6 DVIT

A DVIT (*Digital Vision Touch Technology*) ou a tecnologia de toque por visão digital é uma solução em ambientes verdadeiramente interativos no qual o utilizador pode aceder e manipular os dados num ecrã de elevadas dimensões.

Esta tecnologia da *SMART Technologies* oferece uma precisão sem igual, baixo tempo de resposta e permite a interação com o quadro em tempo real. Recorrendo ao uso de câmaras digitais e algoritmos de reconhecimento sofisticado é possível determinar a posição da extremidade do dedo e distinguir operações como um clique, clique duplo, arrastar ou soltar. Este método apresenta uma solução mais intuitiva para o utilizador operar o quadro e é ideal para apresentações interativas em empresas, bem como no ensino escolar.

2.3 Métodos de projeção

O modo como é exibido o ambiente de trabalho no quadro interativo pode variar consoante a tipologia do sistema. A seguir serão apresentados três métodos de projeção: frontal, traseira e de ecrã LDC ou Plasma.

2.3.1 Projeção frontal

No método de projeção frontal o projetor de vídeo encontra-se em frente ao quadro interativo. Uma desvantagem desta tipologia é que ao interagir com o quadro, o utilizador encontra-se entre o projetor e o quadro interativo, dando origem a uma sombra. Este facto pode ser minimizado pelo utilizador recorrendo ao uso de técnicas de apresentação e posicionamento ou mesmo algumas tecnologias mais recentes apresentam a possibilidade de subir ou descer a posição do quadro, para o adaptar à sua altura.

Com o objetivo de melhorar este tipo de projeção têm sido comercializados quadros interativos de projeção frontal de curta distância como é ilustrado na Figura 5.



Figura 5 – Quadros interativo de projeção frontal de curta distância. Adaptada de [5].

Neste caso o projetor está fisicamente ligado à estrutura do quadro projetando a imagem com um ângulo aproximado de 45°, o que elimina grande parte da sombra, minimizando a principal desvantagem deste tipo de projeção.

2.3.2 Projeção traseira

O método de projeção traseira veio eliminar o problema da sombra gerada pelo posicionamento do utilizador entre o quadro e o projetor. Neste método o projetor está colocado atrás do quadro, logo no caso de o utilizador estar a realizar uma apresentação para uma plateia não necessita de estar de frente para a luz emitida pelo projetor.

A grande desvantagem deste método é o preço. Estes quadros são normalmente mais caros comparando com os de projeção frontal sendo ainda necessária uma maior área para a sua instalação, a não ser que sejam encastrados na parede (Figura 6).



Figura 6 – Quadro interativo de projeção traseira. Adaptada de [6].

2.3.3 Ecrã LCD ou Plasma

Como demonstra a Figura 7, recorrendo ao uso de um ecrã LCD ou Plasma, é possível realizar apresentações interativas, mas este método limita o tipo de tecnologias possíveis de implementar. As mais comuns são infravermelhos e ultrassons.



Figura 7 – Quadro interativo utilizando um ecrã. Adaptada de [7].

2.4 Áreas de utilização

Os quadros interativos são utilizados nas mais diversas áreas, sendo aqui abordadas as que se representam como as maiores consumidoras deste tipo de tecnologia. Inicialmente o uso deste produto era sobretudo no meio empresarial, tendo a dada altura sido reconhecidas as inúmeras capacidades deste tipo de tecnologia como sendo uma ferramenta versátil, altamente inovadora devido às suas características de interatividade e apresentação de conteúdo de uma forma mais eficiente e efetiva, tendo revolucionado todo o método de ensino.

2.4.1 Quadros interativos nas escolas

A introdução de quadros interativos nas escolas em Portugal surgiu ao abrigo do Plano Tecnológico para a Educação na resolução do conselho de ministros n.º 137/2007, de 18 de Setembro [8].

Tendo por objetivo aumentar o leque de equipamentos informáticos nas salas de aula, com vista a permitir práticas pedagógicas mais inovadoras e interativas, esta era uma das metas a atingir por parte do Ministério da Educação, realizar a instalação de um quadro interativo por cada 3 salas de aula até 2010 [9].

Em 19 de Fevereiro de 2010 foi concluída a instalação de 5.613 quadros interativos com projetor de vídeo integrado nas escolas públicas.

Esta tecnologia foi anteriormente implementada em alguns países da União Europeia como o Reino Unido. Num estudo datado de 2010 realizado no Reino Unido [10] declara-se que cem por cento dos coordenadores nas escolas primárias afirmam que este tipo de tecnologia pode ser acedida toda ou a maior parte do tempo, sendo que no secundário e escolas especiais, esses valores são de 84% e 91% respetivamente.

Os professores consideram esta tecnologia adequada ao tipo de ensino que pretendem praticar e permite economizar recursos e tempo. A Figura 8 mostra a percentagem de professores que utiliza este tipo de tecnologia e com que finalidade [10].

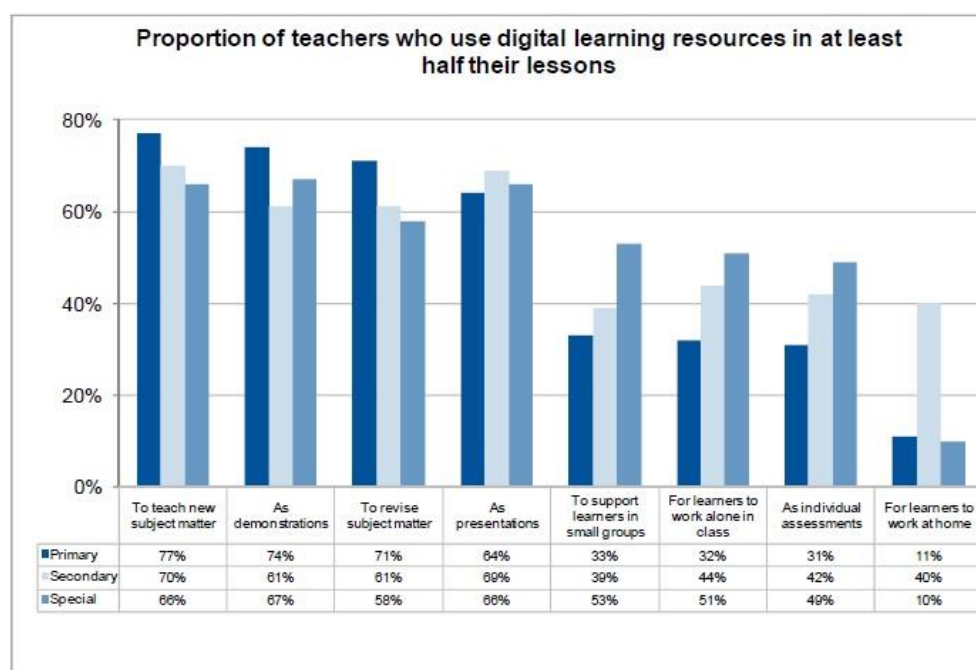


Figura 8 – Percentagem de professores que utiliza os recursos digitais de aprendizagem, pelo menos em metade das aulas. Adaptada de [10].

Os quadros interativos vêm não só substituir os tradicionais quadros a giz ou a marcadores próprios, mas para permitir utilizar um conjunto vasto de funcionalidades e partilhá-las com os alunos.

Muitas das atividades interativas podem ser realizadas com base em funcionalidades simples como arrastar, largar ou corresponder objetos.

Ferramentas de texto permitem adicionar caixas de texto ao ecrã do quadro, sendo possível inserir palavras soltas ou texto. Esta funcionalidade possui as ferramentas normais de um processador de texto para que seja possível alterar o tamanho da letra ou adicionar estilos diferentes.

A maioria do *software* existente para quadros interativos permite o reconhecimento de texto, interpretando o que o utilizador escreve no quadro e transformando-o em texto para que seja possível rever mais tarde.

A utilização desta tecnologia em sala de aula originou o desenvolvimento de um leque vasto de ferramentas para disciplinas específicas como a Matemática. Ferramentas como a régua, compasso ou transferidor são extremamente importantes quando se pretende obter precisão.

Refiram-se ainda outras funcionalidades como permitir partilhar o ambiente de trabalho de um computador e neles executar aplicações educativas, páginas de internet, vídeos,

gravar aulas para poderem ser posteriormente fornecidas aos alunos ou avaliarem o trabalho desenvolvido pelo professor em sala de aula ao longo do ano letivo. A importância das gravações é elevada nos casos de alunos que tenham necessidade de rever a aula ou simplesmente que não tiveram oportunidade de assistir.

Esta tecnologia assumiu um papel determinante na evolução dos métodos de ensino e aprendizagem e por isso os fabricantes de material escolar estão empenhados em responder às necessidades que têm surgido ao nível de educação, e desenvolvendo produtos educativos orientados para o uso de quadros interativos.

2.4.2 Quadros interativos nas empresas

Em meio empresarial, os quadros interativos são fundamentalmente utilizados em escritórios, no âmbito de apresentações em reuniões. São particularmente vantajosos em aplicações interativas e permitem ilustrar e guardar todos os apontamentos retirados para futuramente serem revistos.

Com o avanço tecnológico que se vive hoje em dia, é comum a realização de reuniões, existindo uma separação física dos intervenientes. A necessidade de alguém se deslocar milhares de quilómetros para realizar uma apresentação por vezes não é viável, nem se justifica devido à simplicidade e clareza com que esta tecnologia nos permite realizar uma apresentação.

Esta tecnologia é bastante utilizada em empresas inseridas na área de desenvolvimento de projetos de desenho assistido por computador (CAD) ou *design*.

Devido ao aumento de popularidade dos quadros interativos em empresas, deu-se início ao desenvolvimento de ferramentas que permitem cada vez mais aumentar o leque de funcionalidades para o utilizador, existindo programas para apresentações interativas como o *LibreOffice Impress* ou *MS PowerPoint*, aulas geradas em programas como *Cabris*, *Tabulae*, *Interactive Physics*, ou programas de computador CAD, entre muitos outros.

Concluindo, os quadros interativos permitem no meio empresarial o desenvolvimento de novos métodos de trabalho e comunicação, simples, interativa e possível de se realizar em qualquer lugar e a qualquer hora independentemente da localização de todos os intervenientes.

Capítulo 3

Arquitetura

3.1 Microsoft Kinect

3.1.1 O sensor Kinect

O sensor Kinect é um dispositivo que incorpora diversos componentes eletrónicos que fazem dele um dos sensores de movimento mais utilizados atualmente. A Figura 9 descreve os elementos que o constituem.

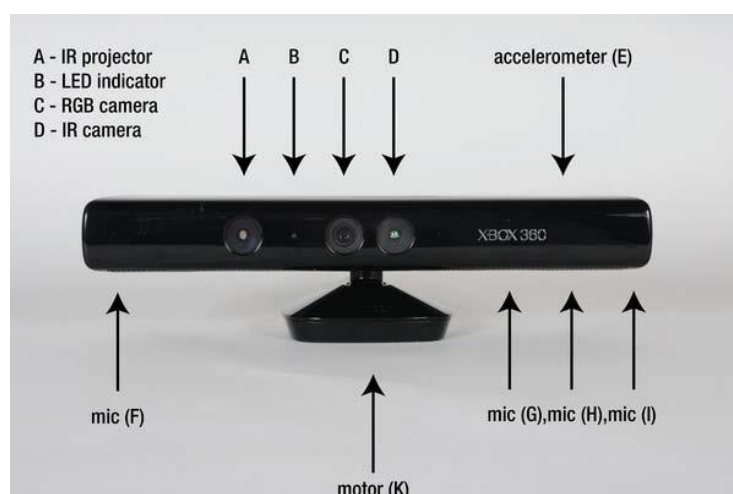


Figura 9 – Identificação de componentes externos do sensor Kinect. Adaptada de [11].

O sensor Kinect é constituído por vários sensores que adquirem a informação do ambiente físico em frente ao sensor.

Existem quatro microfones no sensor Kinect, constituindo um sistema designado *Quadraphonic Sound*. Combinando o referido sistema com um avançado processamento digital, é possível utilizar estes microfones para filtrar o ruído de fundo e detetar a posição relativa do utilizador. Um microfone localiza-se à esquerda identificado com a letra F, e outros três à direita pelas letras G, H e I na Figura 9 [11].

A câmara RGB (Figura 9, C) presente neste equipamento permite adquirir uma imagem RGB, enquanto para a operação de medição de profundidade é utilizado o projetor (Figura

9, A) e câmara (Figura 9, D) de infravermelhos. A câmara de infravermelhos é constituída por um sensor *Complementary Metal–Oxide–Semiconductor* (CMOS) monocromático, que permite adquirir uma imagem tridimensional independente das condições de luz ambiente. O feixe de luz emitido pelo projetor de infravermelhos apresenta uma cor vermelha facilmente identificável. Este passa por uma grade de difração que produz um número de pontos que serão posteriormente captados pela câmara. A intensidade desses pontos traduz uma medida entre o sensor e o objeto a qual se designa por profundidade.

O acelerómetro (Figura 9, E) com 3 eixos encontra-se configurado para uma variação de duas vezes a aceleração devido à gravidade, permitindo obter a orientação atual do sensor.

Um motor identificado pela letra K, consiste num motor de passo que permite movimentar todos os sensores 27° na orientação vertical no sentido positivo e negativo.

Os limites físicos ideais para o bom funcionamento do sensor são descritos na Figura 10.

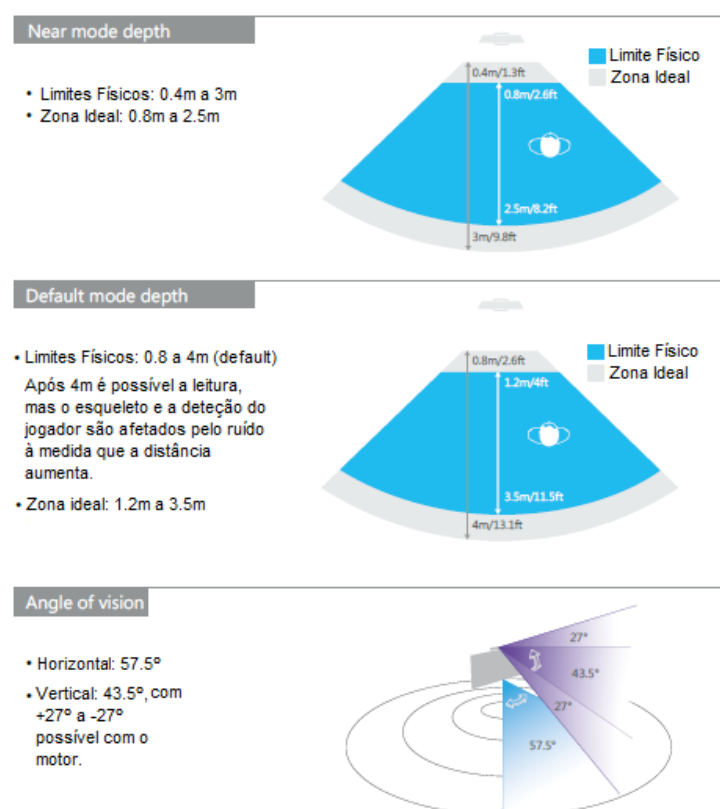


Figura 10 – Limites do sensor Kinect. Adaptada de [12].

3.1.2 SDK (Software Development Kit)

O SDK (*Software Development Kit*) é um conjunto de bibliotecas que permite utilizar a Kinect para o desenvolvimento de aplicações. Com este *software* é possível desenvolver aplicações em *Windows*, *Windows Presentation Foundation* (WPF), *XNA's Not Acronymed* (XNA) entre outros. Como se pode observar na Figura 11, o fluxo de informação ou *streams* são transmitidos para o computador através de uma porta USB.

O NUI API (*Natural User Interface for Kinect Windows API*) coleciona os dados e apresenta-os à camada de aplicação. O SDK também integra o áudio, voz e media API no *Windows* para ser utilizado na aplicação *Microsoft Speech SDK* e o *DirectX Media Object* (DMO) para ser utilizado em aplicações como o *DirectShow* ou *Media Foundation* [13].

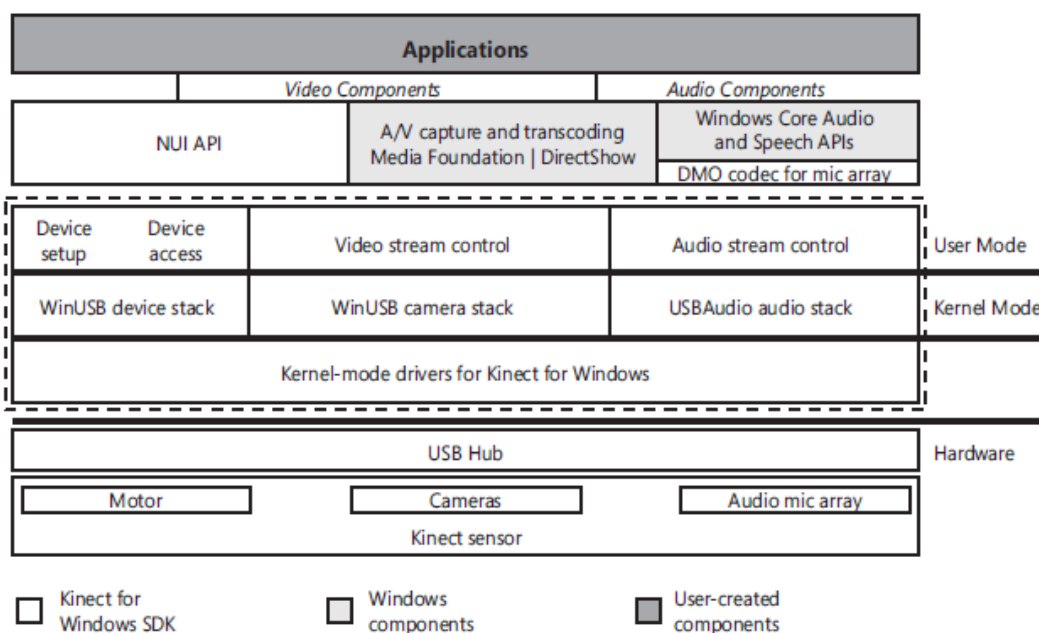


Figura 11 – Arquitetura do SDK Kinect. Adaptada de [13].

Os requisitos de *hardware* para o funcionamento do *Microsoft Kinect SDK* num computador são os seguintes:

- Processador 32 bit (x86) ou 64 bit (x64).
- Processador Dual-core 2.66-GHz ou superior.
- 2 GB RAM.
- Kinect para a Xbox 360.

- USB 2.0 bus.
- Adaptador USB Kinect.

Requisitos de *software*:

- *Microsoft® Visual Studio® 2010 Express* ou outra versão do *Visual Studio 2010*.
- *.NET Framework 4.0*.
- *Microsoft DirectX® SDK* - Junho 2010 ou versão mais recente.
- *Runtime* para *Microsoft DirectX® 9*.

Para reconhecimento de Voz:

- *Microsoft Speech Platform Runtime*, versão 10.2.
- *Microsoft Speech Platform - Software Development Kit*, versão 10.2.
- Kinect for Windows Runtime Language Pack, versão 0.9.

3.1.3 Video Stream

O *video stream* consiste nas frames capturadas pela câmara e pode ser adquirido nos seguintes formatos:

- 640 × 480 - 30fps no formato RGB.
- 1280 × 960 – 12fps no formato RGB.
- 640 × 480 – 15fps no formato YUV.
- 640 × 480 – 15fps no formato Raw YUV.

A imagem RGB encontra-se no formato 32-bit e utiliza um formato linear de cor X8R8G8B8, ou seja, reserva 8 bits para cada cor, num espaço de cor RGB com filtro de cor Bayer, em que cada píxel pode variar entre 0 a 255, inclusive [14].

A imagem YUV é de 16-bit e utiliza um formato linear UYUY-color Bitmap. A utilização deste formato torna o nosso sistema mais eficiente visto que são utilizados menos bits por píxel, logo o driver necessita de menos memória no *buffer*.

3.1.4 Depth Stream

O *depth stream* representa o *stream* de profundidade em que cada píxel corresponde à distância entre o sensor e o objeto mais próximo encontrado. Cada píxel contém a distância expressa em milímetros.

Os formatos possíveis para este *stream* são os seguintes:

- 640 x 480 - 30fps.
- 320 x 240 - 30fps.
- 80 x 60 - 30fps.

Na Figura 12 é possível verificar os pontos resultantes da difração de um feixe de luz emitido pelo projetor de infravermelhos, através dos quais é possível tirar partido da sua informação recorrendo ao uso de técnicas de processamento de imagem que torna possível detetar objetos, esqueletos e a sua distância em relação ao sensor.



Figura 12 – Pontos capturados pela câmara profundidade.

Após o processamento desses dados, é possível realizar um mapeamento dos pontos infravermelhos projetados e os detetados pela câmara, dando origem à imagem da Figura 13.



Figura 13 – Imagem profundidade adquirida pela Kinect.

Na Figura 13 é possível observar diferentes intensidades de cinzento, que correspondem a diferentes profundidades, sendo que as tonalidades se tornam mais escuras à medida que aumenta a distância à *Kinect*. Contudo, existem algumas limitações. Por vezes a presença de um objeto que se encontre demasiado próximo ou afastado do sensor, faz com que este apareça a preto. Isto deve-se ao facto do objeto se encontrar fora dos limites físicos da *Kinect*. A capacidade do local ou objeto refletir a luz infravermelha, a possibilidade de se apresentarem na forma de sombra e não se encontrarem ao alcance do emissor infravermelho são elementos importantes que influenciam os valores de profundidade, visto poderem originar possíveis erros na correta deteção por parte do sensor.

Cada píxel da imagem de profundidade é representado por 16 bits, ver Figura 14, os 13 bits mais significativos representam a profundidade do píxel em milímetros e os três restantes ao índice do utilizador detetado.

0000 1010 1101 0111
Bits de Profundidade Índice do Utilizador

Figura 14 – Bits de um píxel do stream profundidade.

Para este fluxo de informação poder ser utilizado e transformado na imagem designada como imagem de profundidade, é necessário realizar um deslocamento dos 13 bits mais significativos para a direita.

3.1.5 Rastreamento do esqueleto

O NUI API (*Natural User Interface*) utiliza o *stream* de profundidade para detetar a presença de um ou mais utilizadores. A deteção do esqueleto é otimizada para reconhecer um utilizador numa posição frontal ao Kinect. Por isso é necessário que o utilizador não se encontre de lado perante o sensor para que todas as partes do seu corpo se encontrem visíveis.

Podem ser detetadas num máximo seis pessoas, mas apenas poderá ser efetuado o rastreamento do esqueleto a uma ou duas pessoas. O NUI produz o posicionamento de alguns pontos do corpo humano e o conjunto desses pontos é intitulado esqueleto. Esse esqueleto é constituído por 20 pontos, cada um corresponde a uma junta do corpo humano como representado na Figura 15.

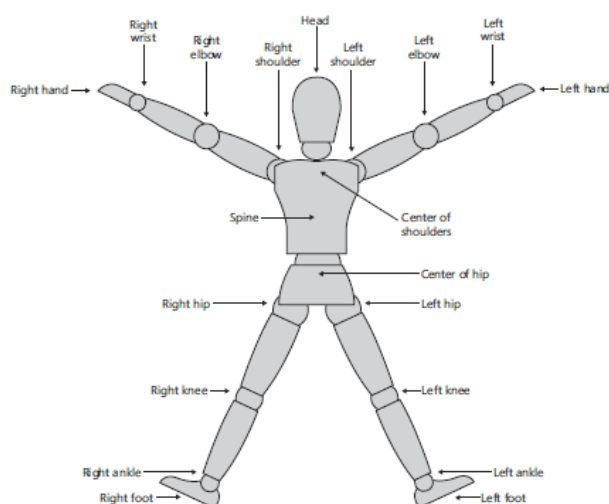


Figura 15 – Conjunto de juntas detetadas pelo sensor Kinect. Adaptada de [13].

Cada junta é definida pelo seu posicionamento no sistema de coordenadas representado na Figura 16.

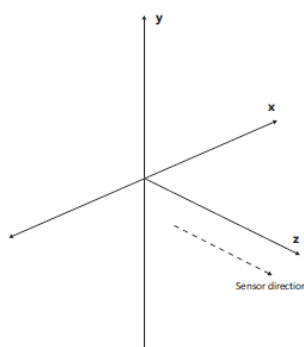


Figura 16 – Sistema de coordenadas das juntas do esqueleto. Adaptada de [13].

O sistema de coordenadas utilizado é definido à volta do sensor, em que o ponto de coordenada ($X=0$, $Y=0$, $Z=0$) corresponde à posição do sensor. As unidades são expressas em metros, ao contrário do valor da profundidade que é expresso em milímetros.

3.2 Emgu CV

O *Emgu CV* é uma plataforma cruzada, entre a *Microsoft.NET* e a biblioteca de processamento de imagens *OpenCV* (*Open Source Computer Vision Library*), o que permite a implementação de funcionalidades do *OpenCV* através de linguagens compatíveis com *.Net* como, *C#*, *VB*, *VC++*, *IronPython* entre outras. Pode ser executado em *Windows*, *Linux*, *Mac OS X*, *iPhone*, *iPad* e dispositivos *Android* entre muitos outros.

A biblioteca *OpenCV* foi desenvolvida pela *Intel* no ano de 2000 e é uma plataforma de livre utilização académica e comercial, para o desenvolvimento de aplicações em visão artificial e processamento de imagem. Possui módulos de processamento de imagem e vídeo, estrutura de dados, álgebra linear, GUI (*Graphical User Interface*) básica com sistemas de janelas independentes, controlo de rato e teclado, além de mais de 350 algoritmos de visão computacional entre os mais importantes podemos identificar os filtros de imagem, calibração de câmara, reconhecimento de objetos, análise estrutural e muitos outros. O seu processamento de imagens em tempo real e a sua eficiência computacional faz com que esta biblioteca seja uma das principais ferramentas utilizadas em visão computacional.

A biblioteca *OpenCV* foi desenvolvida em *C/C++* mas também permite que programadores *Java*, *Python* e *Visual Basic* a incorporem nas suas aplicações.

Esta biblioteca apresenta a seguinte estrutura modular:

Core: Inclui estruturas de dados, operadores e funções básicas de processamento de imagem.

Highgui (High-level GUI and Media I/O): Recursos de interface para o utilizador, codecs de imagem e vídeo, rotinas de captura e conversão de imagens, eventos do rato, como muitos outros.

Imgproc: Algoritmos de processamento de imagem como filtragem, transformações de imagens, conversões do espaço de cor entre outros.

Video: Módulo de análise de vídeo que contém algoritmos como de rastreamento de objetos ou subtração de fundos.

Objdetect: Algoritmos de deteção e reconhecimento de objetos padrão.

Esta biblioteca permitiu neste trabalho realizar operações de segmentação e operação de imagem bem como a extração das características relativas às mãos do utilizador, uma vez que a Kinect não apresenta rastreio dos dedos e não permite a extração de informação dos mesmos.

3.3 Arquitetura do Sistema

Este trabalho pretende desenvolver um interface, capaz de controlar um quadro interativo por gestos, recorrendo ao sensor Kinect.

A maioria dos projetos e aplicações desenvolvidas apresentam uma grande limitação devido à deteção do esqueleto ser um fator determinante para a localização da mão do utilizador. O método desenvolvido não irá recorrer à informação das juntas do esqueleto para permitir ao utilizador operar sem que seja necessário o seu corpo estar dentro da zona visível do sensor, restringindo assim o movimento e deslocação como em operações de escrita num quadro de sala de aula.

Para que isto fosse possível, foi realizado um rastreio e deteção das mãos do utilizador através da deteção da cor da pele, tentando encontrar um método que produza bons resultados e mais abrangente possível a vários ambientes e intensidades luminosas presentes.

Na Figura 17 é proposta uma arquitetura do sistema adotado para atingir o objetivo pretendido. Esta arquitetura é dividida em 5 processos principais: a aquisição de dados, o pré-processamento, a extração das características, classificação de gestos e a atuação.

O primeiro processo é a aquisição de dados pelo Kinect, imagem RGB, o fluxo de informação referente à profundidade e o sensor de voz. Após a aquisição de toda a informação é necessário realizar um pré-processamento das imagens, com objetivo de conseguir extrair as mãos através da cor da pele e profundidade.

A segmentação das mãos pela cor da pele é uma das etapas de maior importância, sem a obtenção de bons resultados seria um grande obstáculo no seguimento do trabalho. O método utilizado para realizar esta segmentação pode definir-se como o cálculo da semelhança que cada píxel tem com um conjunto de pixéis *skin* e *nonskin* e através desse valor de semelhança definir a qual destas categorias pertence.

A imagem de profundidade consiste em filtrar apenas os pixéis que se encontrem dentro de um intervalo de distância que corresponde à zona de atuação onde se pretende detetar os gestos. A combinação entre as duas imagens, profundidade e pela cor da pele dá origem a uma imagem final binarizada onde será possível continuar com o procedimento seguinte.

Após obter uma imagem segmentada das mãos, é possível extrair todas as características correspondentes às mesmas. Esta etapa é importante pois quanto maior for a fiabilidade e qualidade de informação extraída, melhor é o funcionamento da nossa aplicação.

Neste processo as mãos são detetadas e extraídas as suas características, posição, profundidade e número de dedos utilizando funções da biblioteca *Emgu CV*. Estas características irão ser processadas e analisadas sendo efetuada uma correspondência entre uma sequência de pontos de um movimento, a validação de pertencer a um gesto, e se esta última se verificar, a atuação correspondente no sistema.

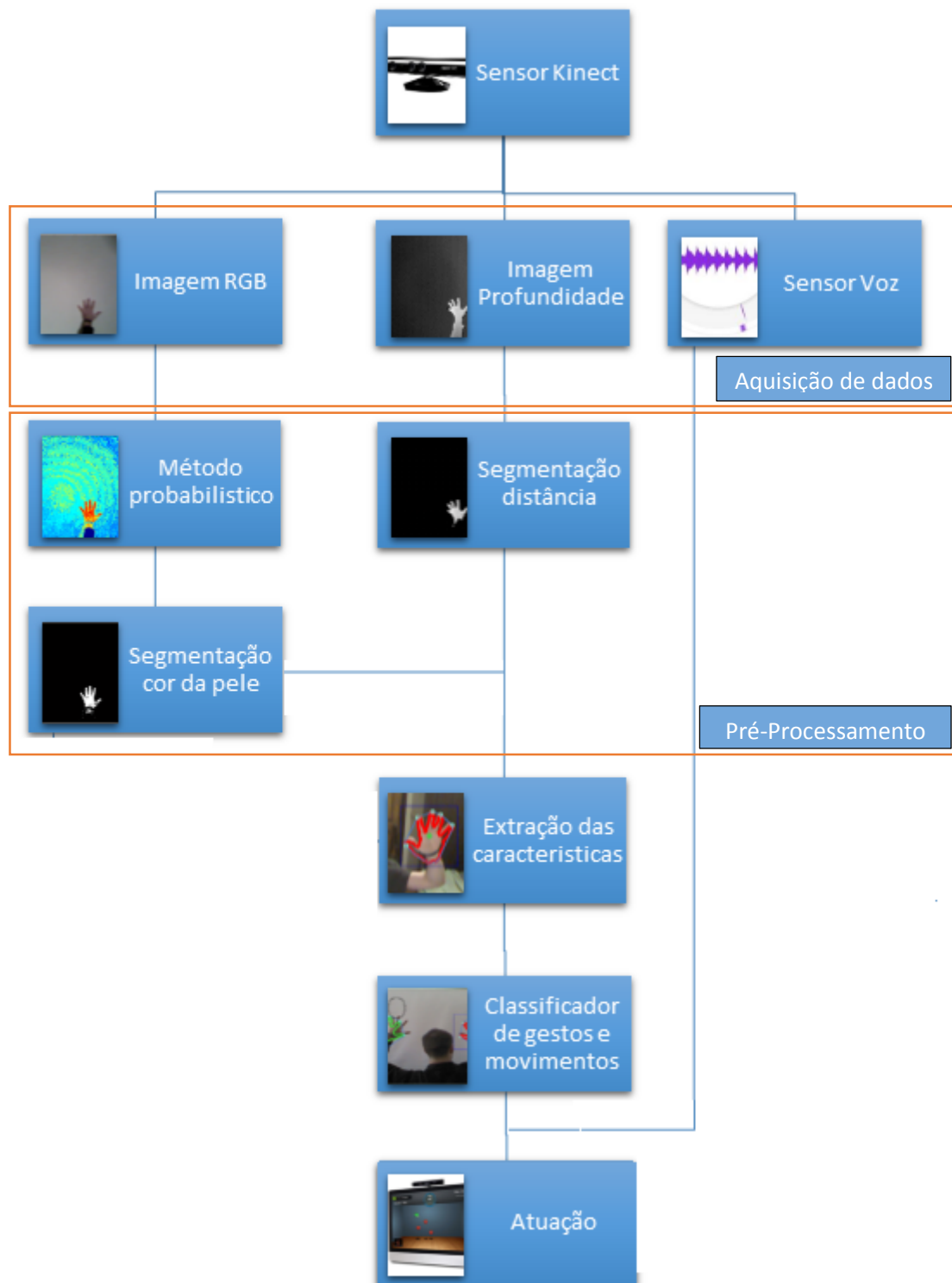


Figura 17 – Diagrama de blocos do sistema a desenvolver.

3.4 Projetos desenvolvidos

O sensor Kinect tem vindo a ser alvo de estudo e implementado nas mais diversificadas áreas de Engenharia, Robótica, Multimédia e Educação devido às suas elevadas capacidades e baixo preço. Esta diversificação deve-se ao fato do Kinect permitir que se retire partido de todas as suas capacidades, como a deteção de juntas do esqueleto humano, áudio e imagem, o que permite o desenvolvimento de projetos com as mais diversas aplicações. Com o desenvolvimento do SDK para *Windows* e de um programa intitulado *Microsoft Kinect Accelerator* [15], a *Microsoft* permite o desenvolvimento de aplicações que utilizem este sensor.

Através deste programa é possível consultar diversas aplicações desenvolvidas das quais se dão maior destaque as seguintes:

3.4.1 Jintronix

O *Jintronix* é um *software* para pessoas com condições neurológicas diminuídas, como por exemplo após um AVC ou lesão cerebral traumática, problemas ortopédicos, ou até para qualquer pessoa que necessite de realizar fisioterapia.

Quando o Pai de Justin Tan's [16] sofreu um acidente vascular cerebral, este acordou no dia seguinte paralisado do lado direito do corpo. Justin deparou-se com as dificuldades como má acessibilidade, falta de empenho e elevado custo na recuperação do seu pai, o que o inspirou a desenvolver um *software* para auxiliar os médicos a minimizar estes problemas aos seus doentes.

Além de ser uma ferramenta de importante auxílio médico, ajuda o profissional de saúde a expandir os seus serviços clínicos tendo o paciente a possibilidade de realizar a fisioterapia no conforto da sua casa, sem ser necessário efetuar deslocações como se pode ver na Figura 18. Este *software* é composto por dois componentes principais: o *JRS Wave*, e o *Portal JRS*.



Figura 18 – Sistema de reabilitação *Jintronix*. Adaptada de [16].

O *JRS Wave* é uma aplicação que pode ser instalada no computador pessoal do utilizador. Contém todas as atividades que o doente necessita para o ajudar a envolver e participar na sua fisioterapia.

O Portal JRS é uma aplicação *web* que permite aos profissionais de saúde obterem acesso direto através do seu *login*, a todas as atividades e acompanhar os pacientes que estão a utilizar o serviço do *JRS Wave*. Através desta aplicação é também possível ao médico prescrever novas atividades e personalizá-los com base nas necessidades do seu paciente.

3.4.2 Skanect

O *Skanect* é uma aplicação que transforma o sensor Kinect num digitalizador 3D de custo reduzido.

A capacidade de reconhecimento 3D por parte do sensor Kinect permitiu o desenvolvimento de possíveis aplicações que retirassem partido desta funcionalidade. Existem no mercado vários digitalizadores 3D capazes de criar modelos de objetos, pessoas ou de locais, mas os custos de aquisição são demasiado elevados para um utilizador comum [17].

Na Figura 19 encontra-se ilustrado a interface gráfica do *software* Skanect.

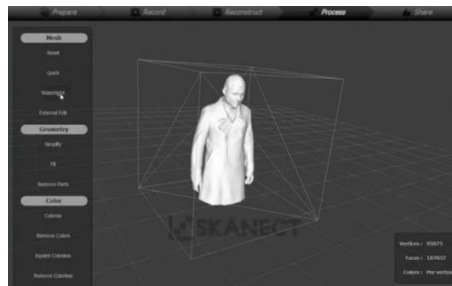


Figura 19 – Exemplo modelo 3D Skanect. Adaptada de [17].

3.4.3 Ikkos

O método *Ikkos* é um sistema de aprendizagem inovador que permite ao utilizador aprender e melhorar a performance de um determinado movimento físico com precisão de uma forma divertida em poucos minutos.

A importância da tecnologia na área de desporto tem vindo a aumentar ao longo do tempo. É essencial que qualquer atleta de alta competição consiga executar com *precisão qualquer movimento que lhe permita atingir níveis de performance mais elevados. Tal não se aplica apenas a atletas, mas a qualquer pessoa que pretenda melhorar o seu desempenho desportivo e recordes pessoais, bem como pacientes de fisioterapia que necessitem de melhorar a sua mobilidade e movimentos.*

O método Ikkos baseia-se em recentes pesquisas da neurociência, utiliza sinais audiovisuais, imagens de vídeo de alta definição combinado com sinais de áudio, especialmente concebidos para ativar áreas do cérebro que são responsáveis por desencadear o movimento muscular.

O processo de aprendizagem Ikkos ativa regiões específicas do cérebro associadas com o movimento do corpo [18].

Na Figura 20 encontra-se o equipamento que nos permite tirar partido desta tecnologia.



Figura 20 – Equipamento *Ikkos*. Adaptada de [18]

3.4.4 Kimetric

Kimetric é uma aplicação inovadora que pretende oferecer aos lojistas, uma análise em tempo real sobre o comportamento e interação que os consumidores têm com os seus produtos, de forma a proporcionar-lhes um método de medir a eficácia das campanhas de marketing nos seus estabelecimentos.

Utilizando apenas o sensor Kinect e o *software Kimetric*, é possível medir o número de clientes, bem como reunir informações sobre a ação e interações dos mesmos em todo o processo, do instante em que se encontram na montra até ao ato de pagamento.

Este sistema não permite o reconhecimento facial pelo que mantém anónima a identidade das pessoas e realiza uma leitura do corpo humano, criando um perfil de consumidor utilizando quatro parâmetros: Sexo, Idade, Altura e Largura. É ainda capaz de reconhecer até quatro estados: Feliz, Irritado, Triste e Surpreso [19].

3.4.5 Picksurelt

Picksurelt é uma aplicação que combina o espaço real e produtos virtuais, utilizando realidade aumentada 3D, permitindo que os seus utilizadores visualizem os produtos em ambiente e escala real.

É um *software* simples de utilizar possibilitando a personalização de cores, materiais e partilha dos produtos nas redes sociais ou internet. Na Figura 21 é possível observar um exemplo da visualização de um produto através da tecnologia *Picksurelt* [20].



Figura 21 – Software *Picksurelt*. Adaptada de [20].

3.4.6 Styku

O *Styku* é uma aplicação de digitalização que permite em apenas 3 segundos, digitalizar o corpo de uma pessoa, permitindo extrair medições precisas com um erro de 0,64cm.

A venda *online* é cada vez mais uma plataforma utilizada por parte dos comerciantes possibilitando cativar um maior número de clientes e vender o seu produto em qualquer parte do mundo.

Este modelo de negócio veio abrir uma nova perspetiva de mercado e sendo a área do vestuário uma das que gere um maior número de clientes foi necessário encontrar uma solução para um problema que resultou deste serviço, nomeadamente, a devolução dos seus produtos devido a erro humano. Este erro humano é fruto de uma má identificação das dimensões e sua correspondência com os tamanhos do fabricante por parte dos consumidores.

O *Styku* pretende minimizar o problema nos processos de medição, garantindo a quem o utiliza medições consistentes e uma maior fiabilidade nas medidas das pessoas e escolha do seu vestuário [21].

3.4.7 Ubi Interactive

A *Ubi Interactive* é uma empresa que desenvolveu em 2013, um projeto em parceria com a *Microsoft* permitindo ao utilizador uma interatividade com um computador apenas recorrendo ao uso de um sensor Kinect [22].

Com os primeiros passos na evolução da interatividade entre o humano e o computador e a necessidade da tecnologia ser adquirida a um custo reduzido, surgiu o desenvolvimento de interfaces que proporcionassem ao sensor Kinect a capacidade de transformar qualquer superfície numa superfície interativa.

Este *software* permite, com o recurso às mãos do utilizador ou um dispositivo do tipo caneta, interagir com uma imagem projetada numa superfície.

3.4.8 Atlas5D

O *Atlas5D* consiste num *software* que oferece uma solução inovadora para segurança e controlo de pessoas dentro de edifícios.

O envelhecimento populacional é hoje em dia um fenómeno universal tanto em países desenvolvidos como também em países do terceiro mundo. Com este fator surge o aumento de pessoas idosas isoladas em casa, que estão sujeitas a várias situações perigosas como uma queda ou um AVC, sendo necessário cada vez mais encontrar soluções simples que melhorem a qualidade de vida das mesmas e ao mesmo tempo se sintam seguras [23].

Este produto foi desenvolvido para otimizar a sua segurança através da leitura dos seus movimentos e com isso determinar possíveis mudanças de estado de saúde, em tempo real, a partir de um dispositivo como o computador, *tablet* ou telemóvel.

Esta é uma aplicação que tem em conta o direito à privacidade do idoso não recorrendo a imagens, vídeos nem registo desta informação. Apenas é efetuado o controlo do posicionamento e postura do utilizador. Quando ocorre algo que não se encontra dentro dos parâmetros normais, é enviado uma notificação para os dispositivos eletrónicos previamente configurados.

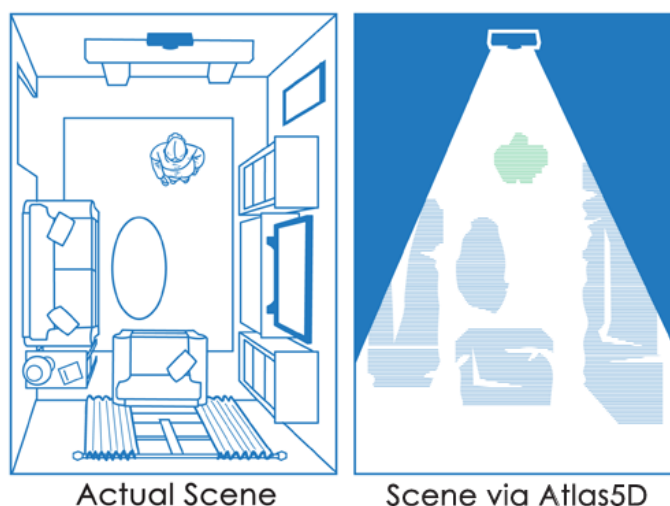


Figura 22 - Software *Atlas5D*. Adaptado de [23]

3.4.9 GestSure Technologies

O *GestSure* é uma solução elegante e rápida que permite o manuseamento de imagens através de gestos simples e intuitivos.

Numa sala de cirurgia é frequente a utilização de ressonâncias magnéticas, radiografias e imagens auxiliares. Visto a dificuldade de manusear um computador dentro de uma sala de cirurgia, o cirurgião vê-se frequentemente obrigado a deslocar-se da sua posição de trabalho para recorrer a esse tipo de informações, fazendo com que perca tempo [24]. A utilização deste *software* pode ser verificada na Figura 23.



Figura 23 – Software GestSure. Adaptada de [24].

3.4.10 Kinect Hand Detection

O *Kinect Hand Detection* é um interface gráfico que realiza o reconhecimento de gestos através do sensor Kinect permitindo a interatividade de um utilizador com as mãos.

Este projeto foi inspirado no filme “*Minority Report*” e utiliza o driver *libfreenect* para interface entre o sensor e o sistema operativo *Linux*. O interface gráfico e o reconhecimento das mãos foram desenvolvidos pelo MIT (*Massachusetts Institute of Technology*) para interagir com o ROS (*Robot Operating System*) desenvolvido por Willow Garage. O *software* de deteção de mãos mostra as habilidades da biblioteca PCL (*Point Cloud Library*), que o MIT tem ajudado a otimizar. Este *software* de deteção de mãos é capaz de distinguir as mãos e dedos numa nuvem de mais de 60.000 pontos a 30 imagens por segundo, o que permite uma interação mais natural e em tempo real. Na Figura 24 verifica-se a deteção das mãos utilizando este interface [25].

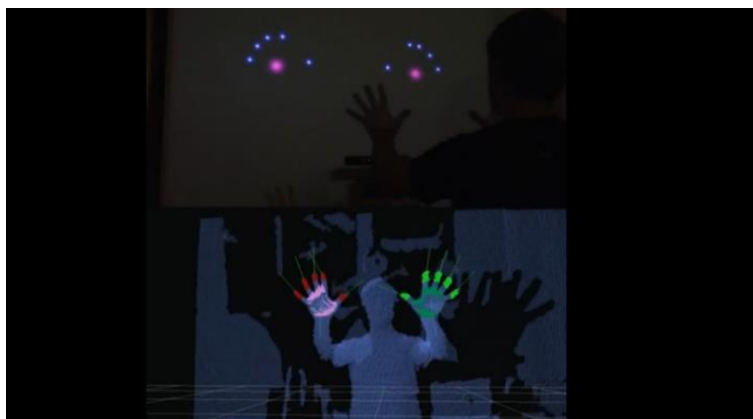


Figura 24 – *Kinect Hand Detection Software*. Adaptada de [25].

Capítulo 4

Desenvolvimento

4.1 Espaços de cor para o reconhecimento da pele

Um píxel pode ser classificado por ser *skin* ou *nonskin* se a sua cor corresponder à pele humana ou não. Esta classificação é tanto melhor quanto maior for a cobertura para todos os tipos de cor de pele (escura, amarelada, branca, entre outras). E também quanto maior for o número de condições de luminosidade consideradas.

Este subcapítulo descreve os espaços de cor, algoritmos de classificação e o método probabilístico estudado.

Apesar de haver vários espaços de cor, muitos deles partilham características semelhantes, têm sido desenvolvidos bastantes estudos para determinar qual o espaço de cor que oferece melhores resultados para a identificação da cor da pele.

Neste trabalho foi abordado os espaços de cor HSV, YCbCr e RGB através de um método probabilístico.

4.1.1 Espaço RGB

O espaço RGB é um dos mais utilizados no processamento de imagens. Composto por três cores primárias o vermelho (R), Verde (G) e azul (B), combinadas entre si a diferentes intensidades, permitem obter milhares de cores.

Numa imagem, cada píxel tem um valor na escala RGB correspondente. Esse valor é representado por três bytes, um por cada componente, o que torna possível obter um valor de intensidade para cada componente de $2^8 = 256$, permitindo obter um conjunto de cores superior a 16,7 milhões.

Sendo este espaço formado pela soma ponderada das três cores, o seu espaço complementar CMY é formado pelo Ciano ($C = 255-R$), Magenta ($M = 255-G$) e Amarelo ($Y = 255-B$) como é apresentado na Figura 25.

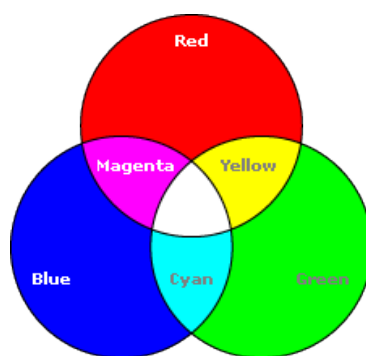


Figura 25 – Sistema de cores aditivo e subtrativo.

Os diferentes níveis de intensidade são normalmente representados numa escala de 0 a 255. Estes valores de intensidade e a sua associação às cores é normalmente ilustrado a partir de um cubo, intitulado Cubo RGB como se verifica na Figura 26.

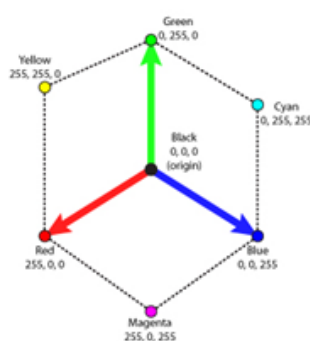


Figura 26 – Cubo RGB.

4.1.2 Espaço HSV

A abreviatura HSV é utilizada para o sistema de cores formado pelos componentes *Hue* (Tonalidade), *Saturation* (Saturação) e *Value* (Valor). É também conhecido por HSB, em que a sigla B significa, *Brightness* (Brilho). Este sistema foi inventado no ano de 1974, por Alvy Ray Smith e consiste numa transformação não-linear do sistema de cores RGB. Além destes sistemas existem outros relacionados como o HSL (L de luminosidade) e o HSI (I de intensidade).

A componente Tonalidade ou Matiz (H), é a medida do comprimento de onda médio da luz que determinado objeto reflete ou emite definindo o tipo de cor do objeto.

A Saturação é a pureza de uma cor. Quanto mais elevado for o valor de saturação, mais pura será essa cor, mas se esta componente apresentar valores inferiores a cor terá uns tons de cinza mais acentuados.

O Valor ou Brilho é o valor que define como o próprio nome indica, o brilho de determinada cor.

Um método utilizado para representar o sistema de cores HSV é o da Figura 27, representado por uma pirâmide hexagonal.

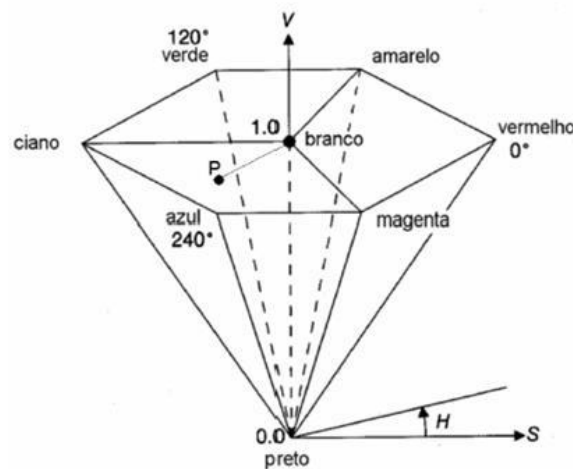


Figura 27 – Modelo hexagonal representativo do espaço HSV.

A altura do hexágono corresponde ao eixo acromático do cubo RGB. No eixo da pirâmide situa-se o parâmetro do Brilho (V), que tem origem no vértice e máximo valor na sua base.

O Valor da Saturação (S) é retirado através da distância de um ponto P, que se encontre em qualquer secção transversal hexagonal da pirâmide até à intersecção do eixo desta pirâmide com a secção hexagonal designado por P_e .

O parâmetro Tonalidade (H) é o ângulo formado pelos segmentos de reta que ligam um ponto P a P_e e o vértice correspondente ao vermelho a P_e , no sentido anti-horário, tendo início nesse mesmo vértice.

Através das Equações 1, 2 e 3 [26] é possível transformar o espaço RGB em HSV.

$$H = \begin{cases} \frac{G - B}{V - \min\{R, G, B\}} \cdot 60 & , \quad \text{if } V = R \text{ e } G \geq B; \\ \left(\frac{B - R}{V - \min\{R, G, B\}} + 2 \right) \cdot 60^\circ, & \text{if } G = V; \\ \left(\frac{R - G}{V - \min\{R, G, B\}} + 4 \right) \cdot 60^\circ, & \text{if } B = V; \\ \left(\frac{R - B}{V - \min\{R, G, B\}} + 5 \right) \cdot 60^\circ, & \text{if } V = R \text{ e } G < B; \end{cases} \quad H \in [0^\circ, 360^\circ[$$

Equação 1 – Cálculo da tonalidade através dos parâmetros RGB.

$$S = \frac{V - \min\{R, G, B\}}{V} \quad S \in [0,1]$$

Equação 2 – Cálculo da saturação através dos parâmetros RGB.

$$V = \max\{R, G, B\} \quad V \in [0,255]$$

Equação 3 – Cálculo do brilho através dos parâmetros RGB.

O espaço HSV apresenta a vantagem de uma fácil percepção sendo possível separar as propriedades da luminância (Y) e da cromaticidade (C), tornando este espaço de cor muito utilizado em trabalhos que requerem a segmentação da cor de pele. Contudo uma desvantagem, é a descontinuidade da matriz e o elevado processamento no tratamento da luminância das imagens, que pode tornar-se prejudicial para a realização de técnicas de detecção de pele [27].

4.1.3 Espaço YCbCr

O espaço de cor YCbCr foi criado para permitir que as emissões dos sistemas de televisão a cores fossem compatíveis com os recetores a preto e branco. Os sistemas SECAM (*Séquence Electronique Couleur avec Mémoire*) e PAL (*Phase Alternating Line*), de 1961, empregam o modelo YCbCr, definido pela norma CCIR 601. Este modelo é muitas vezes designado por YUV em computação gráfica, no contexto de imagens digitais [28].

O parâmetro Y, a luminância representa a informação da imagem em escala de cinza, entre o preto e o branco. O Cb e o Cr são definidos como as diferenças entre as componentes B e R do sistema RGB e a luminância Y.

Ambos os parâmetros se representam por 8 bits, sendo a luminância representada entre os intervalos de valor [16,235] e as componentes da crominância [16,240] [28].

A transformação do espaço RGB para YCbCr é realizado pelas equações 4, 5 e 6 [26].

$$Y = 0,299R + 0,587G + 0,114B$$

Equação 4 – Cálculo da luminância através dos parâmetros RGB.

$$C_b = \frac{B - Y}{2 - (2 \times 0,1145)}$$

Equação 5 – Cálculo da crominância Cb através dos parâmetros RGB.

$$C_R = \frac{R - Y}{2 - (2 \times 0,2989)}$$

Equação 6 – Cálculo da crominância Cr através dos parâmetros RGB.

Utilizando estas fórmulas é possível converter facilmente o espaço RGB em YCbCr possibilitando uma separação explícita dos componentes correspondentes à crominância com a luminância o que torna este espaço de cor, um dos mais utilizados em técnicas de detecção de cor de pele.

4.2 Reconhecimento e detecção das mãos

4.2.1 Introdução

Toda a informação necessária para a o desenvolvimento deste trabalho é obtida através dos métodos de aquisição de dados já anteriormente referidos. As imagens RGB e de profundidade são adquiridas no formato 640x480 a 30fps com a ativação do *Color* e *Depth Stream*. Neste trabalho não foi incluído o *Skeleton Stream*, ou a informação relativa às juntas do esqueleto detetado, porque o objetivo pretendido é a implementação em quadro interativos, onde o corpo do utilizador pode ou não estar completamente visível dentro dos

limites de funcionamento do sensor Kinect o que provocaria certas limitações como o utilizador estar obrigado a um posicionamento mais restrito.

Nos subcapítulos seguintes irão ser descritos todos os processos. Na segmentação pela cor irá ser descrito o método utilizado para detetar a cor da pele. Na segmentação pela profundidade irá ser explicado em que medida esta propriedade irá ser utilizada para definir a zona de atuação.

Em Detecção das mãos e dedos, é descrito como é obtida a informação das características da mão e dedos a partir da imagem segmentada, como tamanho da mão, número de dedos e posicionamento da mesma em referência ao sensor.

Por fim, o classificador de gestos e Atuação onde são descritos os parâmetros que os descrevem e pelos quais os gestos são identificados, assim como a sua atuação no sistema.

4.2.2 Segmentação pela cor da pele

Nos últimos anos tem existido um aumento de interesse no que diz respeito à segmentação da pele, o qual tem por objetivo detetar regiões da pele humana em uma imagem. A segmentação de pele é maioritariamente utilizado para a deteção do rosto [29], análise de movimentos da mão [30] ou filtragem de imagens censuráveis [31].

Nas aplicações referenciadas anteriormente, o espaço de busca de objetos de interesse como a face ou as mãos, pode ser reduzida por meio da deteção da zona de pele. Com esse fim, a segmentação de pele é o método mais eficaz, porque normalmente requer uma menor quantidade de código computacional e é invariável do tipo de pose em que se encontra.

Um dos fatores mais importantes neste trabalho é a deteção da pele humana. Para encontrar uma resposta a este problema foram realizados alguns ensaios com objetivo de selecionar um método que obtivesse uma melhor fiabilidade na deteção da cor da pele. Foram abordados os espaços HSV, YCbCr e RGB utilizando algumas técnicas de classificação de imagem em pixels *skin* e *nonskin*, categorias classificadas com base na cor do píxel e ao qual se tem vindo a aprofundar o seu estudo devido a ótimos resultados que se têm obtido através dos mesmos.

No Apêndice 1, é possível consultar toda a informação relativa aos ensaios realizados no *software Matlab*, com imagens e histogramas analisados.

4.2.2.1 Estudo HSV

Através de um conjunto de algumas imagens com variantes no fator de luminosidade e cor da pele os quais se destacam por obterem uma aproximação ao tipo de imagens que se pretende processar neste trabalho, foi realizado uma análise para o espaço de cor HSV que se encontra no Apêndice 1.

As imagens das Figuras 28 e 29 são referentes à análise da imagem número 4 do Apêndice 1.

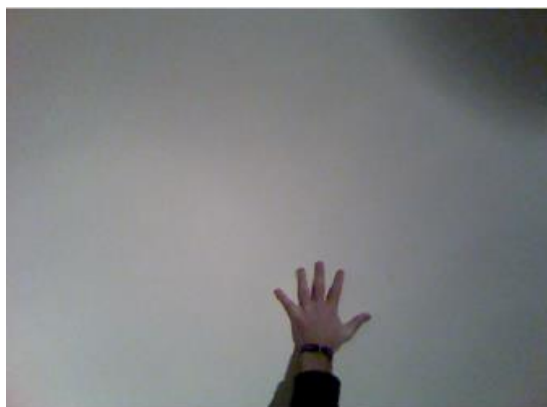


Figura 28 – Imagem RGB.

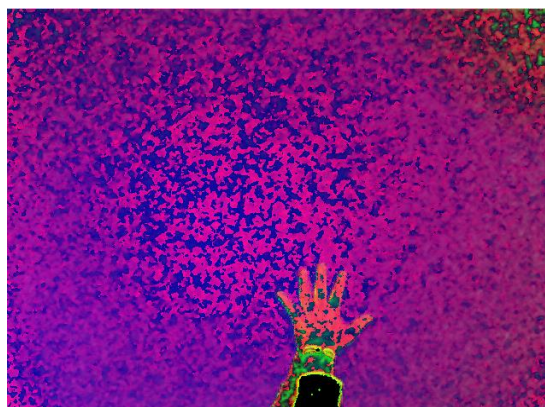


Figura 29 – Imagem HSV.

O objetivo é encontrar um intervalo de cor que permita identificar a pele humana para diferentes tipos de luminosidade e cor de pele. Logo não se achou relevante iniciar a série de ensaios com um fundo de cor introduzindo ruído, mas sim apenas com um fundo branco para que ao analisar o histograma da imagem se tornasse mais fácil a análise e fosse possível definir o intervalo de cor onde a mão se encontra o mais correto possível.

Através do *Matlab* foi inicializado um pequeno programa que se encontra na Figura 30. É efetuada a leitura da imagem RGB da mão, de seguida converteu-se para o espaço de cor para HSV para que seja analisado o histograma de cores sendo este por sua vez analisado e se obtenha o espaço de cor que separa todos os pixéis mão dos pixéis que não o são.


```

1      %% Estudo HSV mão
2      close all;
3      clear all;
4      clc;
5
6      % Mostrar Histograma
7      A=imread('Hand4.png'); %Leitura da imagem número 4
8      B=rgb2hsv(A);          %Conversão RGB para HSV
9      imshow(A);             %Vizualizar imagem RGB
10     imshow(B);              %Vizualizar imagem HSV
11     % Configuração do Histograma
12     [cH, cS, cV, x]=hsvhist(B);
13     hold on;
14     plot(x, cH, '-r');       %Parâmetro H a vermelho
15     plot(x, cS, '-g');       %Parâmetro S a verde
16     plot(x, cV, '-b');       %Parâmetro V a azul

```

Figura 30 – Programa *Matlab* para análise do espaço de cor HSV.

Analisando o histograma da Figura 31 e a realização de alguns ensaios de tentativa e erro é possível identificar os limites de valores para os quais é possível segmentar a imagem de forma a obter apenas os pixéis referentes à cor da pele humana.

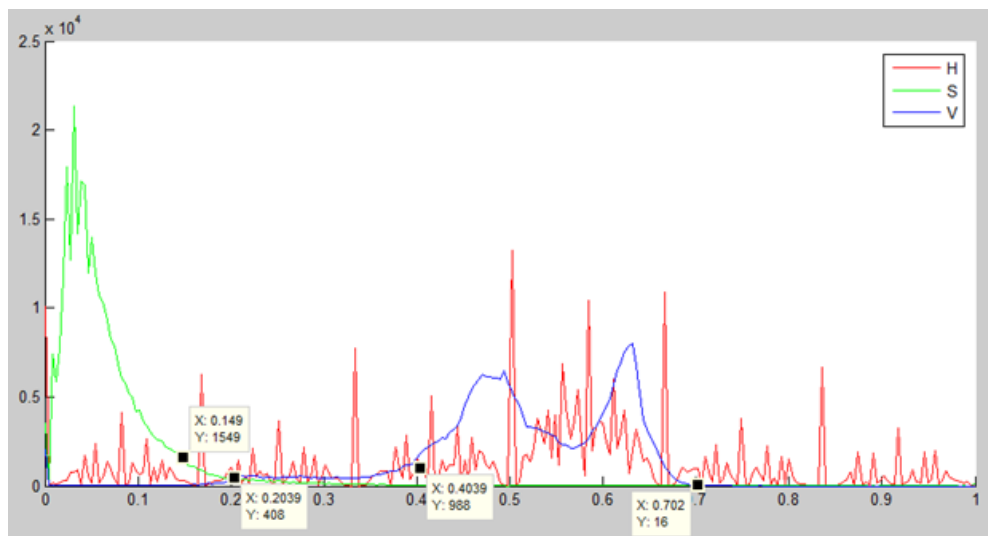


Figura 31 – Histograma HSV.

Após a análise do histograma da Figura 31, é também possível identificar os valores dos parâmetros HSV para a cor da pele. Esta amostragem de valores está representada por dois eixos de coordenadas X e Y, escala de valores dos pixéis no intervalo de 0 a 1 e o número de pixéis respetivamente. O vermelho, verde e azul são as cores escolhidas para a amostragem dos valores correspondentes aos parâmetros H,S e V. Nesta imagem os valores da tonalidade da cor da pele encontram-se entre todos os valores inferiores a 0.4

e superiores a 0.7, o valor da saturação encontra-se situado em todos os valores superiores a 0.15 e o parâmetro do valor ou brilho está entre os valores de 0.2 e 0.7.

A Figura 33 representa a imagem após a segmentação no intervalo de valores definidos anteriormente.



Figura 32 - Imagem da segmentação efetuada pelo espaço de cor HSV.



Figura 33 - Imagem final obtida após a segmentação HSV.

É possível verificar que foi alcançado com sucesso o objetivo de abstrair a mão do resto da imagem. Mas para que seja possível obter um método que produza bons resultados, com variações significativas de luminosidade, foi necessária a realização de mais testes.

Os intervalos de valores das imagens estudadas estão descritos na Tabela 1. A imagem da mão número 6 será tida em conta para a análise seguinte, devido a não se ter conseguido atingir resultados satisfatórios já que foi efetuada em condições de falta de luminosidade extrema e os valores iriam induzir em erro nos valores finais.

Tabela 1 – Resultados dos intervalos de valores HSV.

Número Mão	Hmín	Hmáx	Smín	Smáx	Vmín	Vmáx
1	0.02	0.2	0.2	---	0.55	---
2	0.095	0.95	0.02	---	---	0.87
3	0 0.7	0.1	0.09	---	0.2	0.7
4	0.7	0.4	0.15	---	0.2	0.7
5	0.7	0.4	0.15	---	0.2	0.7

Analisando os resultados é possível verificar que os parâmetros para a identificação dos pixels pertencentes à região da pele são a tonalidade e o brilho, mesmo os seus intervalos de valor tendo uma variação considerável o que torna difícil obter um intervalo mais abrangente. Os valores da saturação não se tornam muito úteis porque atingem valores em todo o intervalo possível.

Para uma melhor análise aos valores de intervalo dos parâmetros e assim ter uma melhor percepção dos mesmos em relação à sua utilidade para detetar a cor da pele, obtiveram-se os gráficos das Figuras 34, 35 e 36.

Cada vértice da figura geométrica, neste caso um pentágono devido a se ter excluído a amostragem da imagem número 6, representa uma imagem. Os valores do parâmetro respetivo variam de 0 a 255 desde a sua origem, o centro da figura e os seus extremos. De cor laranja é representada toda a gama limite máximo de cada parâmetro e a azul claro a gama de valores limite mínimo. Para se considerar uma boa gama de valores, é necessário que a área de azul claro se encontre dentro dos limites inferiores da área a laranja, ou seja, o limite mínimo tem que ser sempre inferior ao limite máximo, ambas as formas se aproximem da figura geométrica, neste caso pentagonal e que a diferença de áreas seja mínima porque se assim for, os intervalos de valores são constantes e mais reduzidos.

A tonalidade é um parâmetro com o qual a divergência de valores não torna possível adquirir um intervalo de valores comum para todas as imagens estudadas como é possível verificar na Figura 34. Isto deve-se principalmente ao fato da cor de pele não ser igual para todas as pessoas.

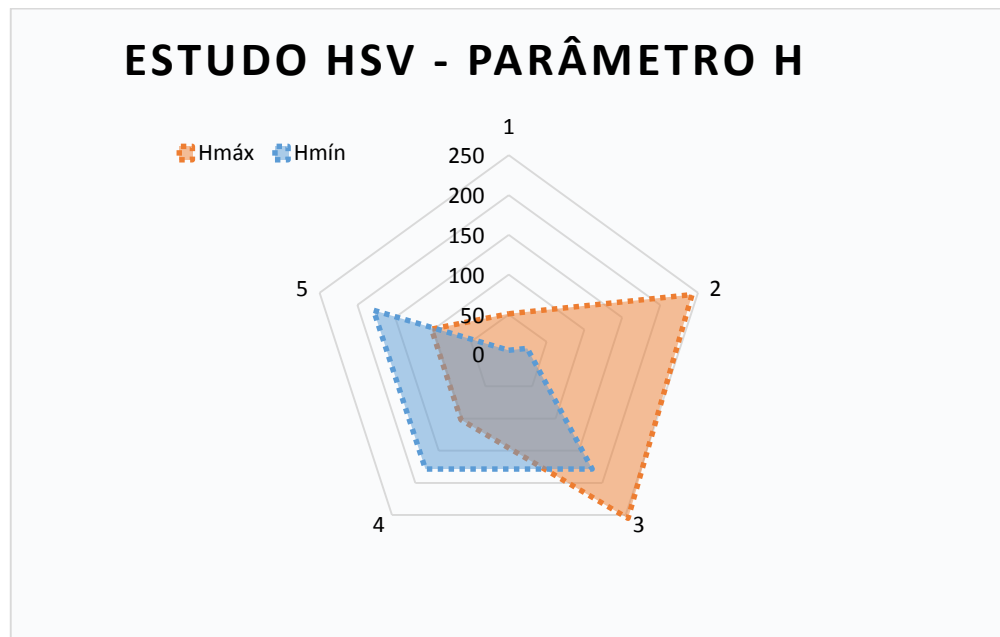


Figura 34 – Gráfico do parâmetro tonalidade.

Na Figura 35 verifica-se que o intervalo de valores do parâmetro da saturação é muito grande o que não nos permite identificar a cor da pele com muita precisão.

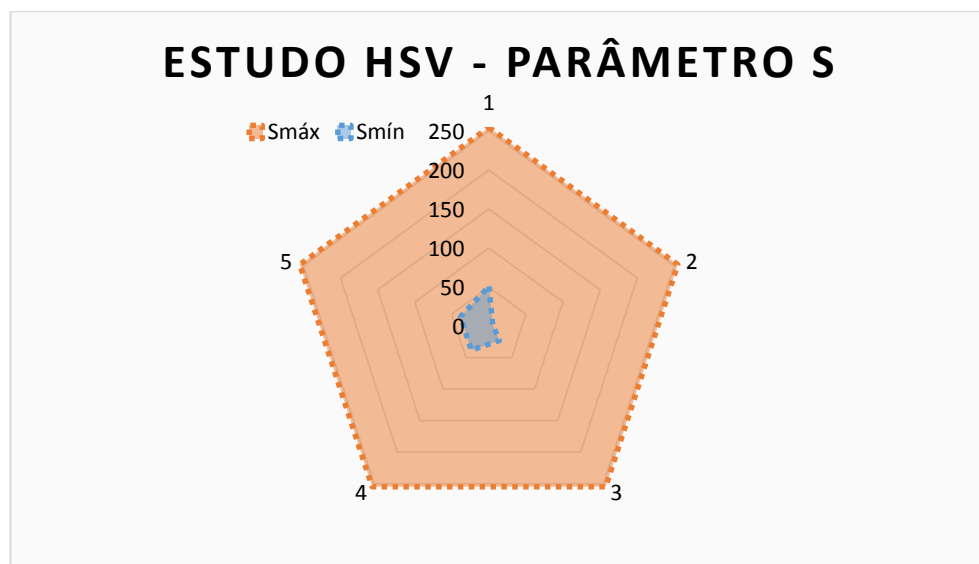


Figura 35 – Gráfico do parâmetro saturação.

O valor ou brilho é para esta situação o parâmetro que apresenta melhores resultados como se verifica na Figura 36.

. Apesar de mesmo assim ter um intervalo de valores considerável, apenas a amostra número 2, apresenta algumas discrepâncias, em que o limite mínimo é um valor em nada parecido com o obtido nas restantes imagens.

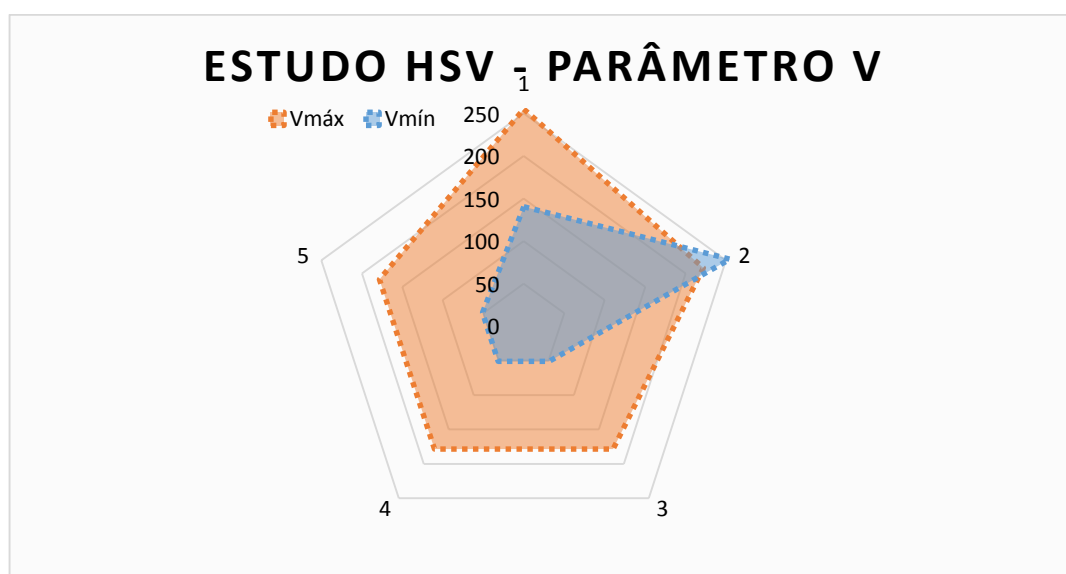


Figura 36 – Gráfico do parâmetro brilho.

Após uma análise cuidadosa é possível dizer que o espaço de cor HSV, apesar de apresentar bons resultados quando efetuada uma calibração rigorosa dos seus parâmetros, dificilmente é um dos melhores métodos para a deteção da pele, visto que é necessário para diferentes tipos de cor de pele e luminosidade, utilizar uma gama de valores bastante diferente. Além destes fatores, este método apresenta algumas desvantagens consideráveis como um elevado custo de processamento na conversão e os pixels com grandes e pequenas intensidades são descartados, tornando os parâmetros H e S bastante instáveis.

4.2.2.2 Estudo YCbCr

O espaço de cor YCbCr também foi um dos elementos em estudo por forma a testar a sua fiabilidade para detetar a cor da pele. Os procedimentos efetuados foram em tudo idênticos ao do HSV, tirando o fato de os parâmetros em estudo serem o Y (luminância), o

Cb e Cr (crominâncias). A mesma imagem referenciada anteriormente irá ser utilizada para uma melhor comparação entre os dois espaços de cor. Nas Figuras 37 e 38 estão representadas a imagem RGB e YCbCr a ser analisadas.



Figura 37 – Imagem RGB.



Figura 38 – Imagem YCbCr.

Na Figura 39 encontra-se representado o histograma da imagem YCbCr, o parâmetro da iluminância Y e das crominâncias Cb e Cr encontram-se identificadas pelas cores, vermelho, verde e azul correspondentemente.

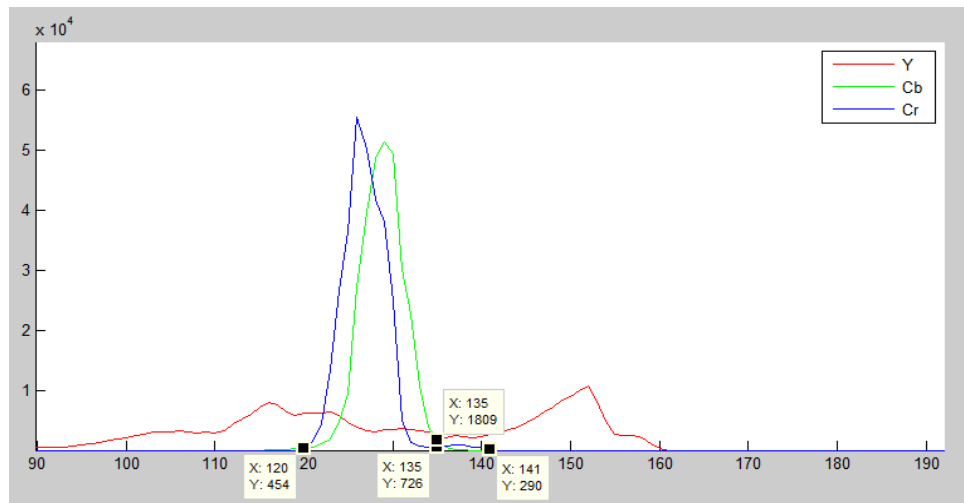


Figura 39 – Histograma YCbCr.

Após analisado o histograma, foi encontrado o intervalo de valores para os parâmetros Y, Cb e Cr para quais apenas os pixels da mão são detetados.

Os valores retirados neste ensaio foi para Cb superior a 120 e inferior a 135 e para Cr superior a 135 e inferior a 141. O parâmetro da iluminância não foi tido em conta para a

distinção dos pixels *skin* e *nonskin* por serem valores demasiados dispersos, que não permitem uma interpretação que nos leve a fazer essa distinção.

Utilizando o intervalo definido obteve-se as seguintes imagens das Figuras 40 e 41.

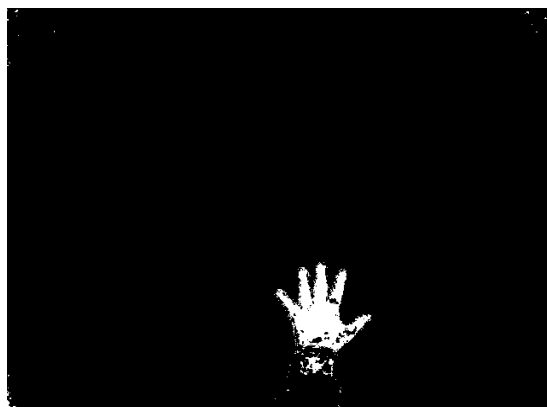


Figura 40 – Imagem da segmentação efetuada pelo espaço de cor YCbCr.

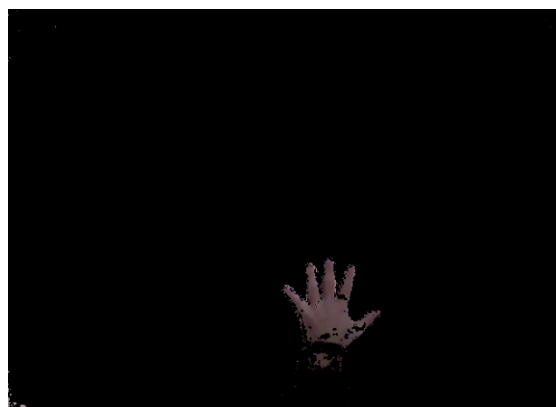


Figura 41 – Imagem final obtida após a segmentação YCbCr.

Pode ser considerado uma boa segmentação da cor da pele, mas como já anteriormente foi referido, para que seja possível obter um método abrangente, é necessário realizar outros ensaios para diferentes tipos de pele e luminosidade.

Os intervalos de valores das imagens estudadas estão descritos na Tabela 2.

Tabela 2 – Resultados dos intervalos de valores YCbCr.

Número Mão	Cbmín	Cbmáx	Crmin	Crmax
1	102	127	133	162
2	110	127	130	160
3	105	133	129	157
4	120	135	135	141
5	120	136	135	141
6	127	129	130	132

A imagem da mão número 6 não será tida em conta para a análise seguinte, devido a não se ter conseguido atingir resultados satisfatórios já que foi efetuada em condições de falta de luminosidade extrema e iriam induzir em erro os valores finais.

Na Figura 42 é possível verificar que o conjunto de valores obtidos para o parâmetro Cb se encontram na mesma ordem de valores, o que o torna possível definir com mais exatidão o intervalo de pixels que correspondem à pele humana.

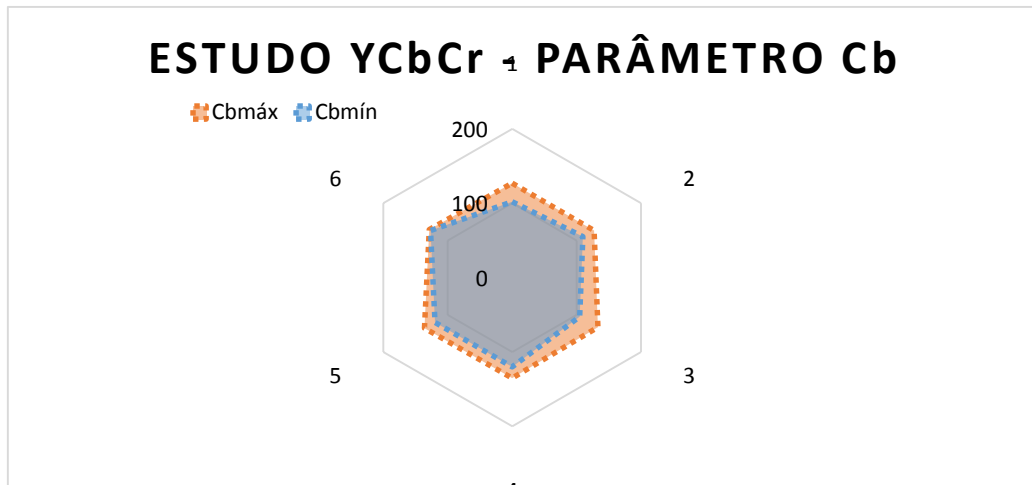


Figura 42 – Gráfico do parâmetro crominância Cb.

A Figura 43, mostra que embora não sejam resultados tão bons como os anteriores também é possível definir um bom intervalo de valores.

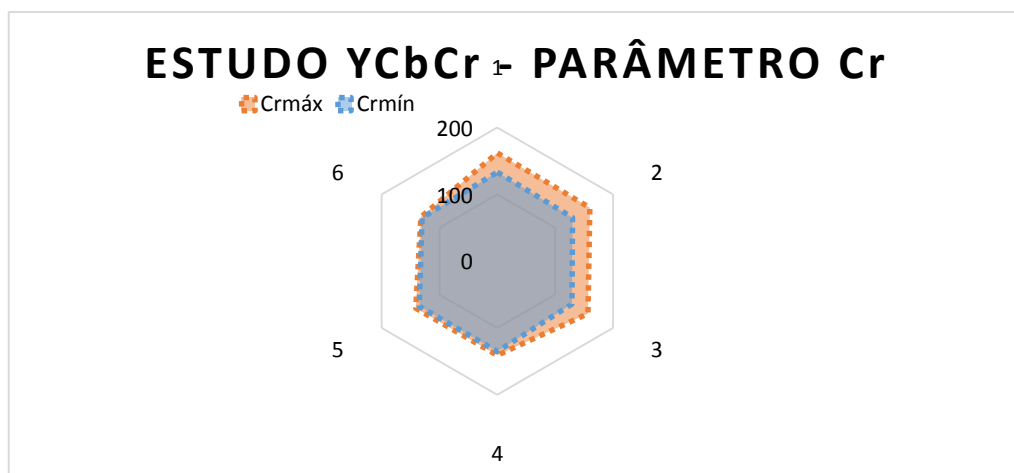


Figura 43 – Gráfico do parâmetro crominância Cr.

Com a obtenção destes resultados, é possível verificar que o espaço de cor YCbCr neste caso e para as imagens utilizadas é o método que apresenta maior fiabilidade e que nos permite obter um intervalo de valores onde é possível realizar uma melhor segmentação de forma a distinguir a cor da pele do resto da imagem.

4.2.2.3 Método probabilístico

Os píxeis referentes à pele humana têm um variado conjunto de cores. Na Figura 44 temos uma representação da gama de cores ao qual pertencem o grupo de pele humana no espaço RGB.

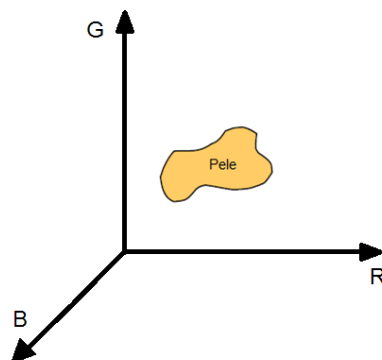


Figura 44 – Representação da gama de cores dos píxeis *skin*. Adaptada de [34].

Um píxel é considerado *skin*, se os seus componentes R,G e B pertencerem à região da pele. Para aprender quais os píxeis que pertencem a esta região é necessário aprender através de exemplos de imagens de pele e daí guardar a informação dos que são *skin* e os *nonskin*. Na Figura 45 encontram-se representados a laranja os píxeis *skin* e a azul os píxeis *nonskin*.

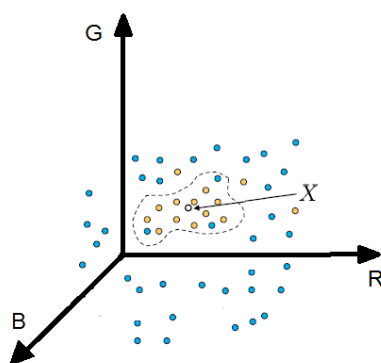


Figura 45 – Representação de píxeis *skin* e *nonskin*. Adaptada de [34].

Como é visível na figura anterior, existem píxeis *skin* que se encontram fora da área considerada pele e outros *nonskin* que também se encontram dentro a gama de cores pele.

Para a correta distinção destes mesmos píxeis recorre-se ao uso de algumas técnicas de classificação.

Na Figura 46 encontra-se um gráfico representativo de um modelo, para determinar a probabilidade de um píxel ser ou não *skin*.

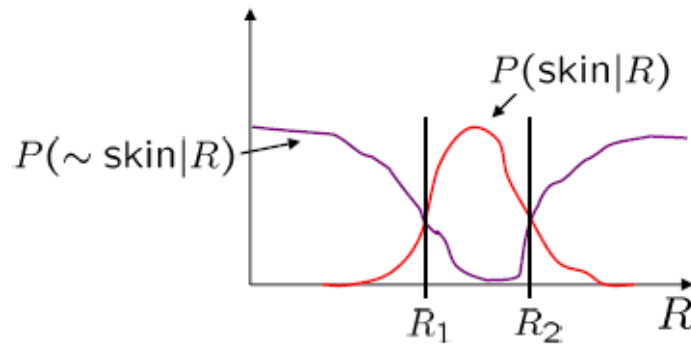


Figura 46 – Representação da probabilidade da componente R ser ou não *skin*. Adaptada de [34].

Sabendo que a probabilidade que um píxel tem de ser *skin* ou *nonskin* é dada através da Equação 7 em que um píxel é considerado *skin* se tiver dentro do limite mínimo R_1 e máximo R_2 que a componente R que a cor da pele pode ter.

$$P(\sim \text{skin}|R) = 1 - P(\text{skin}|R)$$

Equação 7 – Probabilidade de um píxel ser *nonskin*.

Para calcular a probabilidade de ser *skin* ou *nonskin* recorreremos à função de densidade de probabilidade ou PDF (*Probability density function*). Esta função está representada na Figura 47.

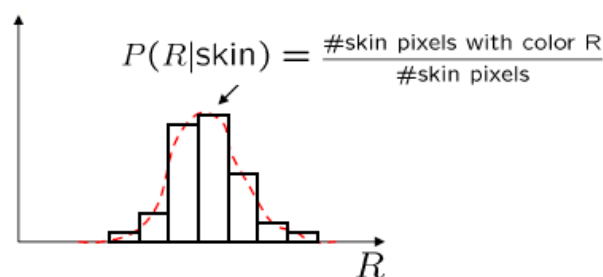


Figura 47 – Representação da função de densidade de probabilidade. Adaptada de [34].

A função de densidade de probabilidade é uma função não negativa utilizada para representar a distribuição de probabilidade caso as variáveis aleatórias discretas sejam contínuas. Esta função trabalha com intervalos infinitesimais e retorna a probabilidade da variável, neste caso componente R, assumir um valor num determinado intervalo.

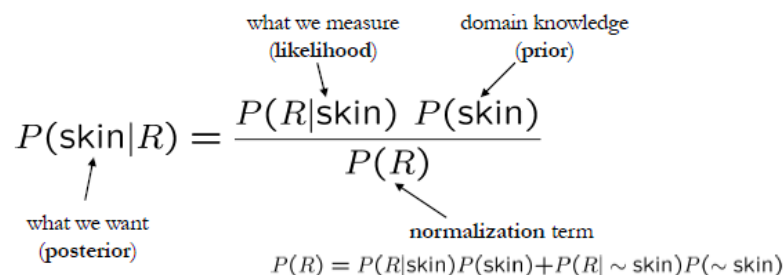
Como o objetivo é saber a probabilidade que determinado valor de R tem em ser *skin* e não o contrário foi utilizado o Teorema de Bayes associado à técnica de histogramas que pode ser construída muito rapidamente mesmo com um elevado número de cálculos em comparação com outros classificadores como o MLP (*Multi-layer perceptron*).

O Teorema de Bayes mostra a relação entre uma probabilidade condicional e a sua inversa como pode ser visto na Equação 8.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Equação 8 – Teorema de Bayes.

Nos termos do nosso problema o teorema de Bayes é utilizado como demonstra a Figura 48.


$$P(\text{skin}|R) = \frac{P(R|\text{skin}) P(\text{skin})}{P(R)}$$

what we want (posterior) what we measure (likelihood) domain knowledge (prior)

normalization term

$$P(R) = P(R|\text{skin})P(\text{skin}) + P(R|\sim \text{skin})P(\sim \text{skin})$$

Figura 48 – Teorema de Bayes aplicado ao nosso problema. Adaptada de [34].

Para este procedimento foram analisados alguns artigos de estudos previamente realizados para determinar que espaço de cor nos oferece uma maior fiabilidade utilizando este método probabilístico. No artigo *Skin segmentation using color pixel classification: analysis and comparison* [32] foram analisados um total de oito vetores de características: RGB, HSV, YCbCr, CIE –Lab, rg normalizada, HS, CbCr e ab. Os primeiros quatro são compostos por vetores de todos os espaços de cor e os últimos quatro apenas em espaços de crominância. Esta análise foi também efetuada em sete diferentes tipos de tamanhos de histogramas de A. Dyadic, 4, 8, 16, 32, 64, 128 e 256.

As curvas ROC (*Receiver Operating Characteristic*) dos 5 vetores de características, com os tamanhos de histograma de 64 e 256 são mostrados na Figura 49 e as taxas de classificação (CRs) dos pontos selecionados nas curvas ROC são apresentados na Tabela 3. Observa-se que para o mesmo tamanho de 256 do histograma, o desempenho da classificação foi quase a mesma para os quatro espaços de cor testada, RGB, HSV, YCbCr, e CIE-Lab.

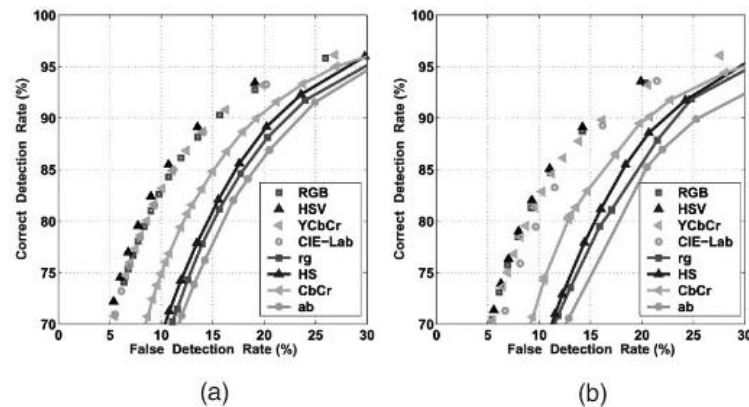


Figura 49 – Curvas ROC (*Receiver Operating Characteristic*) para diferentes representações de cor. (a) Histograma 256. (b) Histograma 64. Adaptada de [32].

Tabela 3 – Taxas de classificação (CRs) de oito espaços de cor (Tamanho Histograma = 256). Adaptada de [32].

	All Channels				Only Chrominance Channels			
Color Representation	RGB	HSV	YCbCr	CIE-Lab	rg	HS	CbCr	ab
FDR = 10%	88.75	88.76	88.46	88.47	84.93	85.46	86.56	84.51
FDR = 15%	86.17	86.19	85.97	85.97	83.92	84.14	84.77	83.45
FDR = 20%	82.97	82.98	82.85	82.83	81.65	81.91	82.11	81.33
CR_{max}	89.79	89.80	89.41	89.43	84.95	85.58	86.75	84.52
99% confidence interval of CR_{max}	[89.74, 89.84]	[89.75, 89.85]	[89.36, 89.46]	[89.38, 89.48]	[84.90, 85.00]	[85.53, 85.63]	[86.70, 86.80]	[84.47, 84.57]

É possível concluir que a classificação do pixel *skin* pode ser feita através da maioria dos espaços de cor, dependendo do formato da imagem de entrada e a necessidade de passos de processamento de imagem subsequentes.

A diferença de desempenho de cada um dos espaços de cor pode ser um efeito do tamanho do histograma. Em tamanhos mais elevados do histograma, a diferença tende a tornar-se menor.

A conclusão deste artigo também concorda com o estudo de M.C. Shin [33] que mediu o *skin* e a sua separabilidade com o *nonskin*, usando métricas derivadas a partir das matrizes classe de dispersão e histogramas.

O método utilizado neste trabalho baseia-se nestes princípios e informação adaptada de um estudo elaborado por Ciarán Ó Conaire [35] que utiliza modelos não-paramétricos baseados em histogramas que foram registados manualmente e treinados os píxeis *skin* e *nonskin*. Foram utilizados um total de 14.985.845 píxeis *skin* e 304.844.751 *nonskin* e criado um histograma RGB para cada um dos tipos de píxeis de tamanho 32x32x32.

Para um píxel qualquer dado, o logaritmo da probabilidade de ser pele é dada pela função seguinte, em que H se refere ao histograma da *skin* e h refere-se ao histograma da *nonskin*.

$$\log \frac{H(R, G, B)}{h(R, G, B)}$$

Equação 9 – Logaritmo da probabilidade de um píxel ser skin.

A mesma imagem referenciada anteriormente irá ser utilizada no formato RGB para o qual este método foi implementado. Na Figura 50 está representada a imagem RGB a ser analisada.

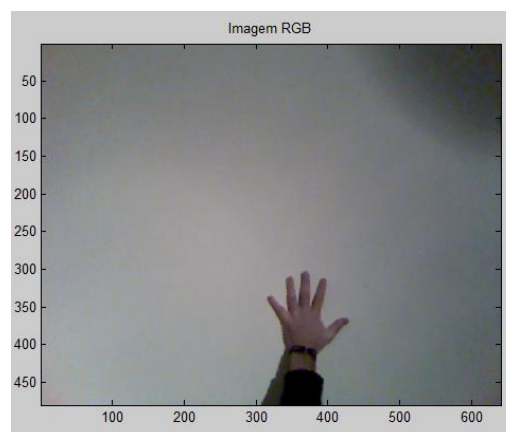


Figura 50 – Imagem RGB.

Na imagem da Figura 51 encontra-se o resultado obtido após ser definido um valor de probabilidade mínimo para que o píxel seja considerado *skin* e na Figura 52 a imagem segmentada determinando um valor mínimo de probabilidade.

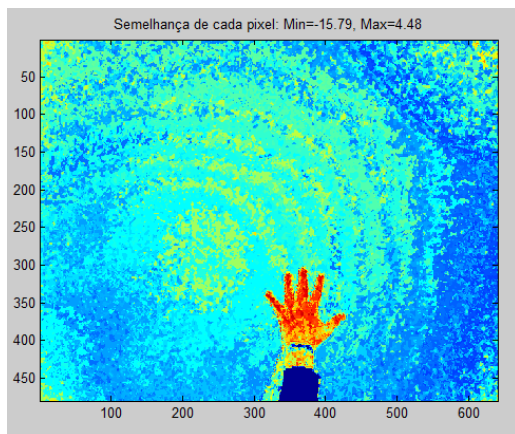


Figura 51 - Imagem método Bayes.

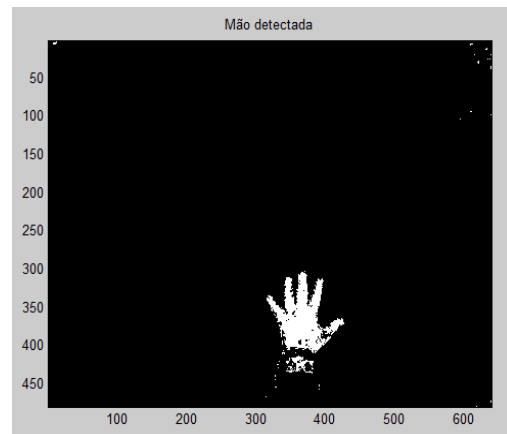


Figura 52 – Imagem segmentada pelo método Bayes.

É possível verificar que nos ensaios realizados com este método, foram obtidos melhores resultados quando comparados com os previamente obtidos dos ensaios HSV ou YCbCr. Consequentemente, este foi o método adotado no desenvolvimento do trabalho para realizar a segmentação das mãos pela cor da pele.

4.2.3 Segmentação pela profundidade

Além da segmentação pela cor da pele, um dos principais fatores para que este equipamento seja tão utilizado deve-se à camara de profundidade e à utilidade que essa informação permite no desenvolvimento de aplicações.

A segmentação pela profundidade é utilizada neste trabalho para criar uma zona de atuação, onde se pretende realizar a deteção das mãos e a leitura dos seus movimentos.

Realizando a aquisição de dados da imagem de profundidade é possível obter apenas os píxeis correspondentes a um intervalo de distância pretendido. A Figura 53 representa a zona de atuação numa vista frontal ao quadro onde os gestos irão ser reconhecidos. O espaço número 1 refere-se às dimensões frontais do quadro interativo e que definem o limite de atuação frontal.

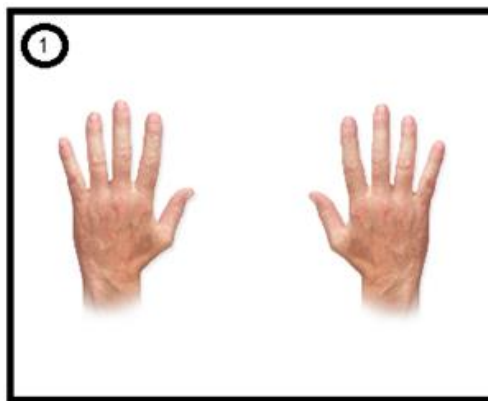


Figura 53 - Representação da zona de atuação na vista frontal

Na Figura 54 o segundo e terceiro espaço correspondem aos limites de atuação numa vista lateral do quadro, sendo que o espaço número três corresponde à zona de atuação e o número 2 à zona onde não é feita a deteção das mãos.

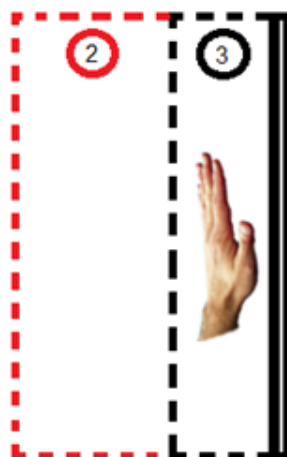


Figura 54 – Representação da zona de atuação na vista lateral.

Este método permite abstrair todos os píxeis que não se encontram na zona de atuação definida anteriormente e com isso é possível realizar uma segmentação da imagem pelo valor da profundidade dos píxeis.

Na Figura 55 temos a imagem de profundidade obtida pelo sensor Kinect em que cada píxel faz corresponder a um valor de profundidade e na Figura 56 a imagem segmentada para valores de profundidade definidos como limites da zona de atuação.

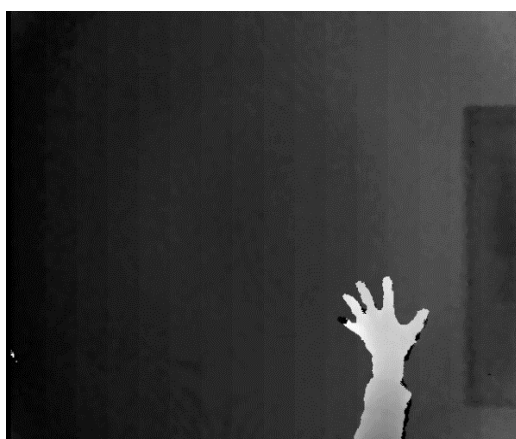


Figura 55 – Imagem profundidade.

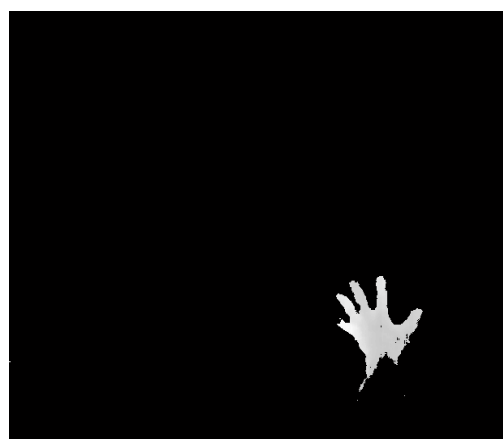


Figura 56 – Imagem da segmentação por profundidade.

Os valores limite foram definidos para a distância que o quadro interativo se encontre a 10cm do sensor. Pretende-se que para estes valores seja possível uma boa segmentação da mão. Após a segmentação essa imagem é binarizada, processada e irão ser extraídas as suas características.

4.2.4 Detecção das mãos e extração das suas características

4.2.4.1 Área da mão

A primeira característica é a área da mão e para isso foram realizados uma série de testes para diferentes distâncias e determinada a sua área em píxeis. Para permitir a identificação da mão independentemente do número de dedos exibidos, é necessário proceder ao cálculo da área com a mão fechada e aberta.

Tabela 4 – Área da mão relativamente à distância em que se encontra.

Teste mão	Distância (mm)	Área (píxeis)
Teste 1 fechada	800	4200
Teste 1 aberta	800	10300
Teste 2 fechada	900	3300
Teste 2 aberta	900	9900
Teste 3 fechada	1000	2900
Teste 3 aberta	1000	8300
Teste 4 fechada	1100	2600
Teste 4 aberta	1100	7400

Após a realização de vários testes foi definida uma área para valores compreendidos entre 2000 e 9000 píxeis. Com isto, pretende-se apenas analisar as características para objetos com áreas compreendidas nesses valores, o que nos permite ter uma melhor robustez na identificação da mão.

4.2.4.2 Centro da mão

Uma característica importante é a localização da palma da mão, informação que irá ser utilizada para o reconhecimento de movimentos. Após a deteção da mão foi definida uma caixa à volta e foi definido que o centro dessa caixa seria utilizado no futuro como o centro da palma da mão como se encontra representado na Figura 57.

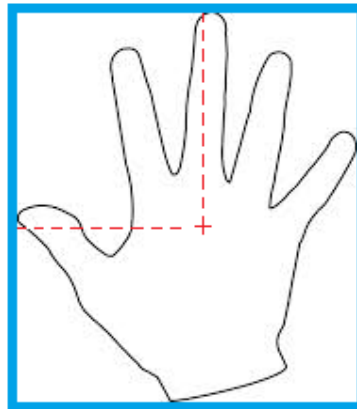


Figura 57 – Representação da caixa e localização da palma da mão.

4.2.4.3 Deteção dos dedos

Segundo Montalvao [36] existem nove pontos característicos que permitem extrair todas as características necessárias para a fase de reconhecimento. Estes pontos encontram-se representados na Figura 58, cinco pontos a vermelho que correspondem às pontas dos dedos e a verde aos vales entre os mesmos.

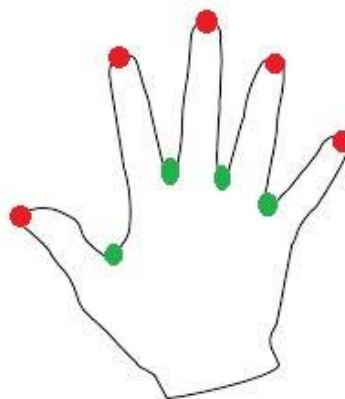


Figura 58 – Representação dos pontos característicos.

A partir do conjunto de pontos anteriormente referidos é possível determinar algumas das características mais importantes como o comprimento da mão, número de dedos, orientação da mão, entre outros.

A biblioteca *Emgu Cv* contém algumas ferramentas de análise de estrutura e descritores de forma, o que permite realizar as seguintes abordagens para extrair a informação relativa aos pontos de interesse. Podemos definir os pontos em dois tipos, *Convexity defects* e *Convex hull*. Na Figura 59 estão representados os *Convexity defects*.

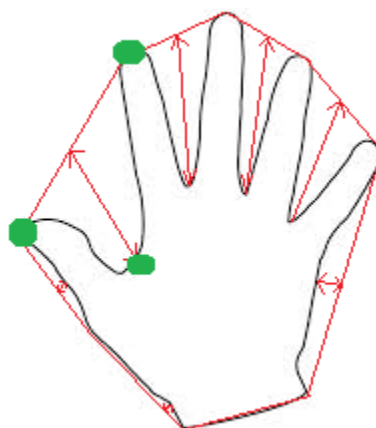


Figura 59 – Representação do cálculo dos *Convexity defects*.

Os *Convexity defects* são as convexidades por defeito no contorno da mão. É realizado o contorno da mão e para cada convexidade é calculado o ponto de início, fim e de profundidade representados com um círculo de cor verde na Figura 59.

Os pontos designados por *Convex hull*, não são mais que o ponto designado por máximo de um determinado conjunto de pontos. A determinação destes pontos de interesse é de cálculo relativamente fácil, porém existem algumas dificuldades como separar as convexidades que realmente interessam e que pertencem aos 9 pontos característicos da mão e os pontos que também são convexidades mas não interessam como se encontra ilustrado na Figura 60.

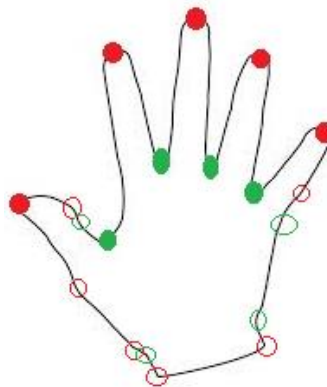


Figura 60 – Representação do conjunto de pontos de interesse e não interesse.

Os pontos designados por *Convex hull* estão identificados a vermelho na Figura 60.

Para a deteção dos pontos correspondentes às pontas dos cinco dedos foram utilizados os pontos *ConvexHull*. No entanto é necessário realizar uma filtragem para identificar os correspondentes às pontas dos dedos e quais os pontos sem interesse.

Considerou-se um ponto O como ilustrado na Figura 61 e definiu-se um sistema de coordenada (X,Y) com o centro nesse mesmo ponto situado na base da mão.

Obtidos todos os pontos convexos, foram calculados todos os segmentos de reta que unem esses mesmos pontos ao Ponto O. Com estes segmentos de reta e definindo um comprimento mínimo entre as pontas dos dedos e a base da mão, é possível identificar com bastante precisão o número de dedos, bem como a sua orientação.

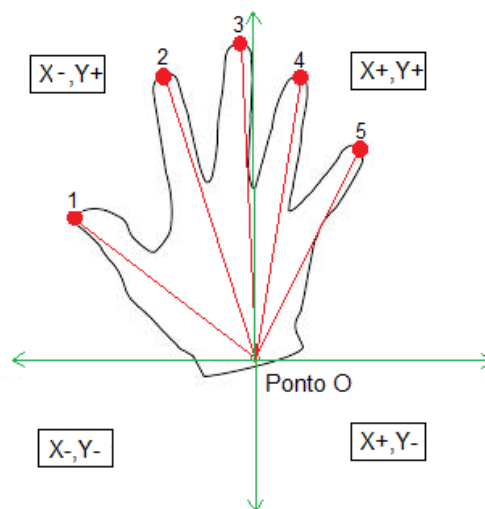


Figura 61 – Método para deteção dos pontos correspondentes às pontas dos dedos.

Na Tabela 5 encontram-se os valores de várias medidas efetuadas para o comprimento dos dedos. Torna-se assim possível comparar os segmentos de reta obtidos anteriormente com estes valores e identificar os dedos da mão. Como verificado através de outros trabalhos que têm por objetivo analisar as características das mãos, o vale entre o indicador e o polegar assume um comportamento bastante inconstante, o que dá origem a uma elevada fonte de erro. A deteção para o dedo polegar identificado por número 5 irá ser feita através do segmento de reta pertencente ao quadrante e com adequado intervalo para o comprimento de reta correspondente.

Tabela 5 – Comprimento dos dedos da mão.

Dedo	Média (mm)	Desvio Padrão	Máximo (mm)	Mínimo (mm)
Nº1	131	13.44	164	95
Nº 2	173	13.47	199	139
Nº 3	185	13.74	215	145
Nº 4	165	13.63	193	109

É de fácil perceção que entre os *Convex hull* existe apenas um dedo logo este também é um fator de filtragem que pode tornar o trabalho mais robusto.

4.2.4.4 Deteção das duas mãos

O método adotado neste trabalho para diferenciar a mão direita da mão esquerda é simples. Se apenas for detetada uma mão, o sistema utiliza a informação dessa mão para a identificação dos gestos, se forem detetadas duas mãos, é realizado um cálculo adicional para verificar as posições das mãos e consoante as suas localizações em relação ao plano de atuação é definido se pertence à mão esquerda ou direita.

Não existe para já um método cem por cento eficaz para realizar este tipo de verificação, vai depender sempre da robustez que o programador der ao seu trabalho. Fatores como, o utilizador estar de costas ou com os braços cruzados, posições não normais, podem resultar numa má identificação da mão esquerda ou direita.

4.3 Classificador de gestos e movimentos

Dan Saffer é um designer de interação e autor de quatro livros, *Designing Devices* (2011), *Designing Gestural Interfaces* (2008), *Designing for Interaction* (2006, 2009) e o seu último livro, *Microinteractions*. No seu livro *Designing Gestural Interfaces*, define alguns dos indicadores mais importantes para a identificação de gestos:

Presença – A presença de um utilizador é o atributo mais básico de todos. Algum objeto ou utilizador tem que estar presente para fazer com que um determinado gesto possa despoletar uma interação. Em alguns sistemas, a simples presença é o suficiente para que possa ser desencadeado algum tipo de ação ou evento na aplicação. No caso do Kinect, a simples presença de um utilizador desencadeia um evento de *tracking* do esqueleto.

Devido a não ser utilizado neste trabalho a informação relativa ao esqueleto, a presença do utilizador é realizada com a deteção das mãos. No máximo é detetado um utilizador, ou seja, um par de mãos.

Duração – Todos os gestos têm um tempo de realização e estes podem ser efetuados de uma forma mais rápida ou lenta. A duração de um gesto vai depender muito do tipo de aplicação em causa. Para alguns interfaces que requerem apenas simples ações como pressionar um botão, mantê-lo pressionado ou deslizar o dedo sobre uma barra como é bastante comum nos telemóveis para o desbloquear, o tempo é um atributo menos importante, mas para outros casos torna-se essencial ter uma duração mais pequena de forma a obter uma reação mais rápida por parte da aplicação. Na área dos jogos principalmente mas também em outros tipos de aplicação, este atributo apresenta uma maior relevância, porque é essencial realizar uma atuação no sistema de uma forma bastante rápida e precisa.

A duração do gesto é um fator tido em conta neste trabalho, sendo um parâmetro classificador dos gestos. Como o trabalho se destina à interatividade com um quadro interativo o tempo de execução dos gestos é de grande importância, para permitir uma reação e atuação rápida por parte do utilizador.

Posição – A posição permite ao sistema saber a localização onde o gesto está a ser realizado. Na maioria das aplicações o gesto é definido estabelecendo uma relação entre a sua localização num sistema de coordenadas XY e calculando a sua alteração. Alguns gestos também incorporam o eixo Z relativo à profundidade. A posição pode estar relacionada com as dimensões do utilizador como a sua altura ou de valores exatos relacionando-a com o meio ou ambiente onde está inserida a aplicação. No trabalho presente esta posição refere-se à posição da palma mão, pois é ela que permite identificar a presença de um utilizador e realizar os gestos que permitem atuar no sistema.

Movimento – O movimento permite ao sistema reconhecer se o utilizador se encontra parado num determinado local ou se encontra em movimento alterando a sua posição e neste caso, que tipo de deslocamento efetua, de frente para trás, ou para os lados. Este parâmetro não foi utilizado no trabalho, visto que a deteção do utilizador é realizada através da presença das mãos e as mesmas são utilizadas para realizar os gestos, logo, não é possível a partir delas saber se o utilizador se encontra em movimento ou parado.

Pressão – Esta característica é apenas associada a gestos que pretendem ser executados em dispositivos *touchscreen* e indica o grau de pressão que o utilizador está exercer sobre o mesmo. No caso da nossa aplicação este atributo não irá ser utilizado.

Dimensão – A partir da altura e largura da área em que o gesto ocorre é possível determinar o comprimento desse gesto. Utilizando esse comprimento é possível classificar os gestos relativamente ao seu comprimento mínimo e máximo e definir o valor desse intervalo para que determinado gesto seja validado. A dimensão do gesto é uma das características utilizadas para classificar os gestos neste trabalho.

Orientação – A orientação diz respeito à direção em que o utilizador está em relação ao dispositivo aquando da execução de um gesto. A orientação é caracterizada como o ângulo do utilizador em relação ao sensor e determinado através de um conjunto de pontos fixos.

Inclusão de objetos – Existem interfaces gestuais que permitem a utilização de objetos por parte dos utilizadores. Estes podem ser interpretados como uma extensão do corpo do utilizador ou em sistemas mais complexos reconhecidos e a sua utilização pode ser

permitida no contexto da aplicação. Um exemplo de objetos bastante comuns são as canetas, que permitem ao utilizador realizar todas as operações necessárias para a interação entre o utilizador e a aplicação. Neste trabalho não se irá recorrer à utilização de objetos pelo que todos os gestos serão identificados através das mãos do utilizador.

Número de pontos de contacto e combinações – Uma característica importante em certas aplicações *touchscreen* é o *multi-touch*, permitindo que o utilizador recorra a mais do que um dedo ou mão em simultâneo para controlar a aplicação. É possível ainda utilizar uma combinação de gestos, associados entre si, resultando num evento específico. Um exemplo bastante comum e desenvolvido neste trabalho é o de utilizar ambas as mãos para redimensionar uma imagem ou rodar.

Sequência – Uma sequência de gestos consiste em dois gestos diferentes independentes um do outro que podem resultar numa ação diferente quando executados numa determinada sequência. Este método implica um interface mais complexo e com uma capacidade de efetuar os registos dos gestos anteriores.

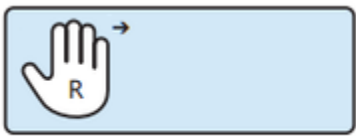


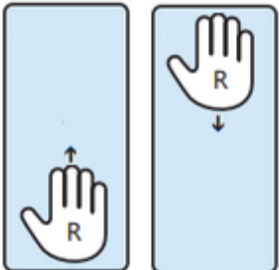
Número de participantes – Dependendo do tipo de aplicação que se pretende desenvolver o número de utilizadores em simultâneo pode ser superior. Esta característica permite a interação em atividades colaborativas e é já muito utilizada nos vídeo jogos mais recentes. O número de participantes neste trabalho é de apenas um elemento, ainda não foi desenvolvido um método que permita identificar as características das mãos de um utilizador.

4.4 Atuação de gestos

Tendo o gesto sido detetado, falta realizar a sua correspondente atuação no sistema. Uma das mais utilizadas ferramentas para realizar apresentações informáticas é o programa *Microsoft PowerPoint*. Este *software* permite a criação, edição e exibição de apresentações gráficas.

A cada gesto detetado irá ser correspondida uma atuação no *software* e na Tabela 6 encontram-se descritos os gestos e a sua correspondente atuação.

Tabela 6 – Lista de gestos e atuações.

 <p>Recuo</p>	<p>Ação de <i>slide</i> anterior</p> <p>Movimento realizado com a mão direita, no sentido horizontal da esquerda para a direita.</p>
 <p>Avanço</p>	<p>Ação de <i>slide</i> seguinte</p> <p>Movimento realizado com a mão esquerda, no sentido horizontal da esquerda para a direita.</p>
 <p>Ampliar</p>	<p>Ação de ampliar <i>slide</i></p> <p>Movimento horizontal com as duas mãos, no sentido horizontal com progressão de dentro para fora.</p>
 <p>Reduzir</p>	<p>Ação de reduzir <i>slide</i></p> <p>Movimento horizontal com as duas mãos, no sentido horizontal com progressão de fora para dentro.</p>
 <p>Cima Baixo</p>	<p>Ação de <i>slide</i> inicial e último <i>slide</i>.</p> <p>Movimentos horizontais, no sentido vertical de cima para baixo ou de baixo para cima.</p>

Existem alguns problemas associados à interação com gestos em aplicações. A memorização dos gestos torna-se difícil se forem de número bastante elevado e não forem intuitivos. Para utilizadores com menor experiência pode vir a tornar-se uma dificuldade.

Uma solução que torna possível diminuir os erros de deteção de gestos e diminuir as dificuldades por parte dos utilizadores facilitando a interação com a aplicação é introduzir comandos por voz. Os comandos por voz permitem efetuar várias tarefas, como inicialização, seleção de opções, o desencadeamento de eventos e sair da aplicação, auxiliando o utilizador e o modo como ele interage com o sistema.

Foram efetuados os seguintes comandos por voz descritos na Tabela 7.

Tabela 7 – Descrição de tipos de comandos por voz.

Comandos	Operação	Descrição
“Forward” “Back” “Up” “Down” “Zoom In” “Zoom Out”	Gestos	Associado a cada gesto irá haver um comando por voz que efetue a mesma operação.
“Paint”	Iniciar o modo desenho	Ativação do modo desenhar na imagem.
“End”	Sair do modo desenho	Sair do modo desenho na imagem.
“Close”	Desligar	Comando para desligar programa.

Capítulo 5

Aplicação e análise de resultados

5.1 Introdução

Neste capítulo irá ser abordado, o funcionamento dos procedimentos desenvolvidos no interface para permitir realizar o reconhecimento de gestos, bem como uma análise aos resultados obtidos.

Irá iniciar-se com a aquisição das imagens RGB, profundidade e como é efetuada.

De seguida será descrito o resultado da correção do erro de *offset* do sensor Kinect recorrendo ao método de calibração desenvolvido neste trabalho.

A segmentação das imagens pela cor da pele e profundidade, bem como as operações morfológicas utilizadas para realizar o seu tratamento também são apresentadas.

O procedimento para realizar a extração das características das mãos e a sua deteção é descrito detalhadamente.

São apresentadas as classes e descrito o seu funcionamento para o reconhecimento dos gestos e os parâmetros que permitem fazer a identificação dos gestos.

A ação que determinado gesto permite realizar no interface encontra-se no subcapítulo controlo.

A escrita, desenho interativo e comando por voz são os últimos temas abordados neste capítulo.

5.2 Aquisição de dados

O primeiro procedimento para o desenvolvimento da aplicação é a inicialização e a receção dos dados referentes à imagem RGB e dos dados de profundidade. Na Figura 62 encontra-se o código referente à inicialização dos *streams* da imagem RGB e profundidade com o formato 640x480 a 30fps.

```
//-----  
kinectSensor.ColorStream.Enable(ColorImageFormat.RgbResolution640x480Fps30); //Inicializa o ColorStream para receber a imagem da camara RGB  
kinectSensor.DepthStream.Enable(DepthImageFormat.Resolution640x480Fps30); //Inicializa o DepthStream para receber a imagem de profundidade  
kinectSensor.Start(); //Inicialização o Sensor Kinect  
  
kinectSensor.ColorFrameReady += kinectRuntime_ColorFrameReady; //Evento para a captura do frame da camara RGB  
kinectSensor.DepthFrameReady += kinectSensor_DepthFrameReady; //Evento para a captura da imagem profundidade
```

Figura 62 – Inicialização dos eventos para receção das imagens.

Na Figura 63 estão os eventos para a captura dos *frames*.

```
//-----  
//----- Evento captura imagem Profundidade-----  
//-----  
private void kinectSensor_DepthFrameReady(object sender, DepthImageFrameReadyEventArgs e){..  
//-----  
//----- Evento captura imagem BGR-----  
//-----  
private void kinectRuntime_ColorFrameReady(object sender, ColorImageFrameReadyEventArgs e){..  
//-----  
//----- Extrair todos os contornos dos dedo-----  
//-----
```

Figura 63 – Eventos para receção das imagens.

No evento “*kinectRuntime_ColorFrameReady*” é realizado a receção da imagem RGB, armazenada na variável *frame* e posteriormente convertida essa imagem para o formato de imagem da biblioteca *Emgu CV* como se encontra representado na Figura 64.

```
//-----  
//----- Evento captura imagem BGR-----  
//-----  
private void kinectRuntime_ColorFrameReady(object sender, ColorImageFrameReadyEventArgs e)  
{  
  
    using (var frame = e.OpenColorImageFrame()) // Guarda a imagem Color na variavel frame  
    {  
        if (frame == null) // Se não obter imagem retorna  
            return;  
        // Obter a imagem do frame RGB  
        Bitmap a = frame.ToBitmap(); // Conversão frame para Bitmap  
        currentFrame = new Image<Bgr, Byte>(a); // Variavel Imagem BGR actual  
        frameBGR = new Image<Bgr, Double>(a); // Cópia da Imagem BGR actual com valores em Double  
    }  
}
```

Figura 64 – Aquisição da imagem RGB.

O evento para o qual se irá proceder ao tratamento de dados referentes à informação de profundidade encontra-se descrito na Figura 65.

```
//-----  
//----- Evento captura imagem Profundidade-----  
//-----  
private void kinectSensor_DepthFrameReady(object sender, DepthImageFrameReadyEventArgs e)  
{  
    using (DepthImageFrame depthFrame = e.OpenDepthImageFrame())  
    {  
        if (depthFrame == null)  
        {  
            return;  
        }  
  
        depthFrame.CopyPixelDataTo(depthPixels);  
  
        depth32 = new byte[depthFrame.PixelDataLength * 4];  
  
        this.GetColorPixelDataWithDistance(depthPixels);  
  
        video3.Source = BitmapSource.Create(depthFrame.Width, depthFrame.Height, 96, 96, PixelFormats.Bgr32, null, depth32, depthFrame.Width * 4);  
    }  
}
```

Figura 65 – Aquisição da imagem profundidade.

Após adquirida a imagem de profundidade, é criado um *array* com nome *depth32* que vai servir para armazenar os valores da profundidade e ser utilizado no método “*GetColorPixelDataWithDistance*” que irá servir para filtrar os píxeis que se encontram a uma distância pré-definida na zona de atuação, já referido anteriormente, onde se espera a realização dos gestos.

5.3 Correção do offset

Com a necessidade de trabalhar com a imagem RGB e de profundidade em comum, surge um problema que é a existência de um *offset* vertical e horizontal como é possível observar nas Figura 66 e Figura 67.

Um artigo publicado por K. Khoshelham [37] refere os principais fatores envolvidos no erro do sensor Kinect: o erro do próprio sensor, erro na medição da intensidade de luz infravermelha ou erro relacionado com a propriedade refletora da luz infravermelha na superfície do objeto.

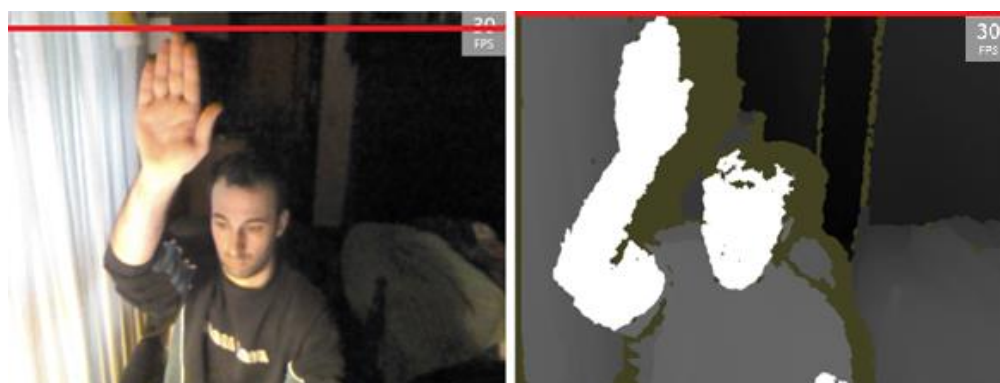


Figura 66 – Offset vertical entre a imagem RGB e infravermelhos.

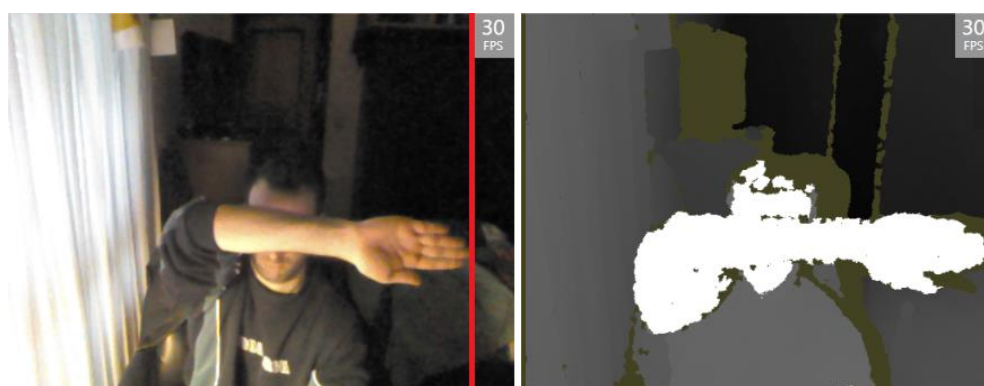


Figura 67 – Offset horizontal entre a imagem RGB e infravermelhos.

Para um dispositivo que funciona corretamente os erros do sensor referem-se a calibragem incorreta e imprecisão na mediação de disparidades. Com a calibração inadequada surge o erro da estimativa dos parâmetros da calibração o que leva a um erro sistemático nas coordenadas dos pontos individuais de cada objeto. Este erro pode ser reduzido efetuando-se uma calibração. Os erros de medição imprecisa das disparidades dentro do algoritmo de correlação e erros de arredondamento durante o resultado de normalização, são erros mais suscetíveis de natureza aleatória.

Os erros provenientes da medição da intensidade da luz infravermelha estão principalmente relacionados com as condições de iluminação e da geometria da imagem. Na presença de uma intensidade de luz forte, os feixes de laser aparecem em baixo contraste na imagem de infravermelhos, o que pode levar a uma lacuna na nuvem de pontos resultante.

A geometria da imagem inclui a distância ao objeto e a orientação da superfície do objeto em relação ao sensor. A distância de trabalho é de acordo com as especificações entre

0,4m a 4m e o erro aleatório da medição da profundidade aumenta com o aumento da distância ao sensor. Além disso, dependendo da geometria da imagem, parte do espaço capturado pode sofrer de oclusão devido à existência de sombras.

As propriedades da superfície do objeto são de elevada importância pois o seu impacto na medição de pontos infravermelhos é elevado. Em superfícies lisas e brilhantes que apareçam expostas, a sua medição é dificultada resultando numa lacuna na nuvem de pontos na imagem de profundidade.

Com o objetivo de minimizar este erro e conseguir que ambas as imagens se encontrem centradas é necessária a realização de uma calibração que permita calcular os deslocamentos entre elas e corrigi-los. O método encontrado para solucionar este problema foi a realização de um deslocamento no posicionamento dos dados de profundidade.

Selecionando a opção de calibração, é pedido ao utilizador que coloque a sua mão numa posição pré-definida pela pelo programador, neste caso optou-se pelo centro da imagem RGB como é apresentado na Figura 68.

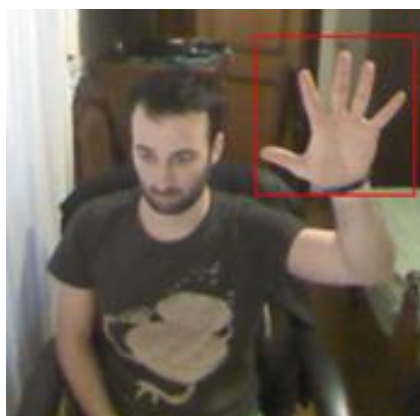


Figura 68 – Modo de calibração a calcular erro.

Após a mão do utilizador se encontrar localizada dentro do quadrado vermelho, é necessário que aí se mantenha durante um determinado período de tempo. Determinada a posição do centro do objeto mão na imagem de profundidade e conhecida a posição definida na imagem RGB, é calculado o deslocamento nos eixos vertical e horizontal necessário para que seja possível realizar o seu alinhamento. A calibração do interface finaliza ao fim de poucos segundos com a alteração da cor da caixa para verde como se pode observar na Figura 69.

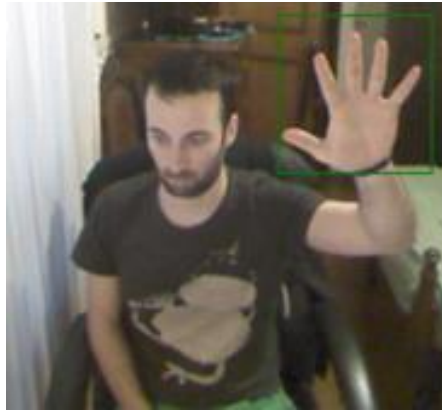


Figura 69 – Modo de calibração finalizado.

5.4 Segmentação pela cor da pele

O processo pelo qual se realiza a segmentação das mãos pela cor da pele inicia-se após a aquisição da imagem do sensor. Para dar início a esta fase é necessário aceder ao ficheiro *skinmodel.bin* que contém os histogramas dos píxeis *skin* e *nonskin* fruto do estudo de Ciarán Ó Conaire [35], que é utilizado neste trabalho no processo de segmentação. Na Figura 70 apresenta-se um fluxograma simplificado dos passos efetuados neste processo.

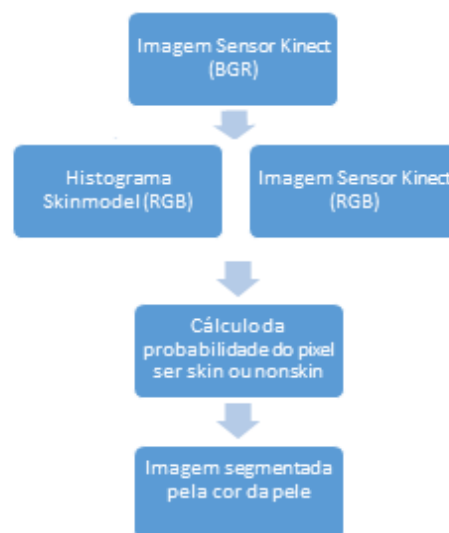


Figura 70 – Fluxograma do processo de segmentação pela cor da pele.

Este estudo foi desenvolvido em *MatLab*, sendo necessário converter a imagem adquirida pelo sensor no formato BGR para RGB no formato demonstrado na Figura 71 para a correta utilização dos histogramas.

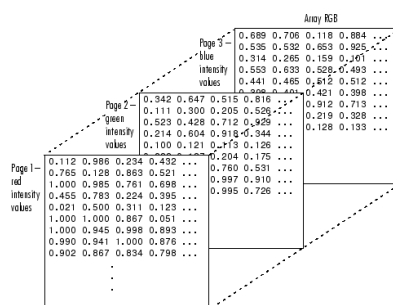


Figura 71 – Exemplo de matriz multidimensional utilizada. Adaptado de [38].

Após colocar os valores dos píxeis da imagem no formato necessário, é feita a verificação pormenorizada de cada píxel e o valor de semelhança que este tem com os píxeis *skin* e *nonskin* dos histogramas.

A deteção dos píxeis *skin* vai depender muito do valor de semelhança pré-definido e para o qual se considera que a partir dele um determinado valor é um píxel correspondente à pele humana, este valor foi selecionado após a realização de vários testes em diferentes tipos de luminosidade e no qual se verificaram melhores resultados.

A partir deste processo é possível concluir que são obtidos ótimos resultados, também já anteriormente verificados no capítulo 4. A sua implementação é simples e torna possível segmentar uma imagem pela cor da pele. Na Figura 72 encontra-se um exemplo de uma imagem RGB ao qual se pretende detetar os píxeis *skin*.



Figura 72 – Imagem RGB sensor Kinect.

A Figura 73 mostra o resultado do método anteriormente explicado e desenvolvido para realizar a deteção dos píxeis *skin* neste trabalho.



Figura 73 – Imagem do resultado da segmentação pela cor da pele.

Como se pode verificar na imagem da Figura 73 é necessário recorrer à utilização de operações morfológicas para melhorar a qualidade da imagem binarizada e seja possível a sua utilização na extração das características da mão. A biblioteca *Emgu CV* é uma biblioteca que oferece ferramentas no campo da área da visão computacional muito importantes, com um leque de operações morfológicas e filtros que permitem melhorar substancialmente a imagem. Alguns exemplos das operações utilizadas estão descritos na Tabela 8.

Tabela 8 – Operações morfológicas do *Emgu CV*.

Operações Morfológicas
<p><i>Dilate:</i> Dilata um elemento estrutural específico.</p> <pre>Bitmap b = detect.ToBitmap(); Image<Bgr, Byte> skin = new Image<Bgr, Byte>(b); StructuringElementEx trat_2 = new StructuringElementEx(4, 4, 2, 2, Emgu.CV.CvEnum.CV_ELEMENT_SHAPE.CV_SHAPE_RECT); CvInvoke.cvDilate(skin, skin, trat_2, 1);</pre>
<p><i>Erode:</i> Elimina um determinado elemento estrutural definido.</p> <pre>Bitmap b = detect.ToBitmap(); Image<Bgr, Byte> skin = new Image<Bgr, Byte>(b); StructuringElementEx trat_1 = new StructuringElementEx(4, 4, 2, 2, Emgu.CV.CvEnum.CV_ELEMENT_SHAPE.CV_SHAPE_RECT); CvInvoke.cvErode(skin, skin, trat_1, 1);</pre>

Smooth: Suaviza uma imagem de várias formas.

```
System.Drawing.Size kernelSize;  
double kernelHorizontalDeviation;  
double kernelVerticalDeviation;  
Bitmap b = detect.ToBitmap();  
Image<Bgr, Byte> skin = new Image<Bgr, Byte>(b);  
CvInvoke.cvSmooth(skinmodel, skinmodel, SMOOTH_TYPE.CV_GAUSSIAN, kernelSize.Width, kernelSize.Height, kernelHorizontalDeviation, kernelVerticalDeviation);
```

Recorrendo à utilização destas importantes ferramentas foi possível atingir uma imagem considerada de boa qualidade como na Figura 74.



Figura 74 – Imagem binarizada após tratamento.

Para uma melhor otimização deste processo de deteção da mão, foi importante relacionar a imagem da segmentação pela cor da pele e de profundidade. Como é possível verificar na Figura 75 é possível detetar todas as partes expostas do corpo, logo torna-se necessário de algum modo conseguir separar as mãos do resto do corpo.



Figura 75 – Imagem segmentação por pele.

Este problema foi resolvido segmentando a imagem de profundidade. Partindo do princípio que apenas as mãos estarão dentro da zona de interesse onde se pretende realizar a deteção dos gestos, é possível abstrair todo o meio envolvente. Recorrendo à utilização de uma operação lógica, resultante da intersecção das duas imagens segmentadas, profundidade e cor da pele obtém-se o resultado da Figura 77.



Figura 76 - Imagem antes da operação lógica.



Figura 77 - Imagem final após operação lógica.

5.5 Extração das características da mão

Para se extrair as características das mãos foram utilizadas ferramentas da biblioteca *Emgu CV* como já referido, as características que se pretendem identificar são a posição da mão e o número de dedos. Para isso foi realizada uma classe chamada *ExtractContourAndHull* que faz uso das operações *findContours*, *contourArea*, *convexHull* e *convexityDefects* que pertencem ao grupo de funções de análise estrutural e descritores de forma da biblioteca *Emgu CV* como referido no capítulo 4.

Na primeira fase deste processo, como se encontra demonstrado na Figura 78 é possível identificar a área onde a mão detetada se encontra, através de um quadrado de cor azul-escuro.

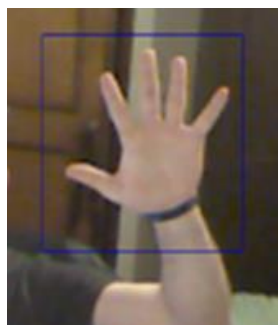


Figura 78 – Área localizada da mão.

Identificada a zona onde se localiza a mão, é possível obter os contornos da mão, como se observa na Figura 79 com uma cor vermelha. Este contorno é obtido a partir da imagem binarizada, resultante da conjugação da imagem de profundidade com a de RGB, logo podem surgir algumas deformações em relação ao aspeto real da mão.



Figura 79 – Contorno da mão.

A fase seguinte consistiu em identificar o centro da palma da mão e o contorno convexo da mesma, característica muito importante pois a partir dela irá ser possível identificar a posição da mão. Como se observa na Figura 80, a verde a localização do centro da mão e a azul claro o seu contorno convexo.

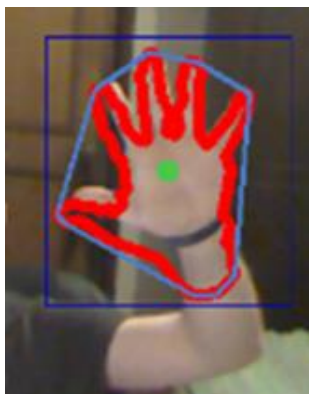


Figura 80 – Centro e contorno convexo da mão.

Através da definição de métricas em parceria com a análise dos defeitos existentes na área convexa da mão é possível a identificação dos dedos e a sua localização. Este é um ponto importante na extração das características e o sucesso do método desenvolvido pode ser verificado na Figura 81 em que os dedos se encontram identificados por círculos nas suas extremidades.

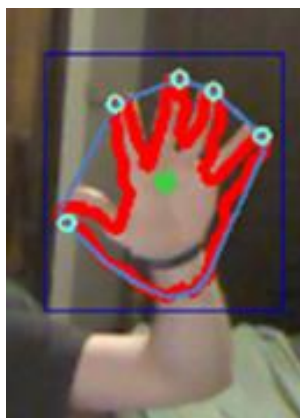


Figura 81 – Identificação dos dedos da mão.

A partir deste momento é possível determinar se a mão se encontra fechada ou aberta consoante o número de dedos expostos e a sua localização. A partir da posição da mão e número de dedos, foi desenvolvido um leque diverso de gestos os quais podem ser associados ao comando de diversas ações.

5.6 Detecção das duas mãos

O mesmo processo desenvolvido para a identificação das características de uma mão foi utilizado no caso da deteção das duas mãos em simultâneo. Cada mão é tratada como um objeto diferente e identificadas as suas características.

Na Figura 82 é possível verificar o resultado do método desenvolvido. As duas mãos estão identificadas corretamente, a cor vermelha é utilizada para representar a mão direita e a cor verde a mão esquerda.

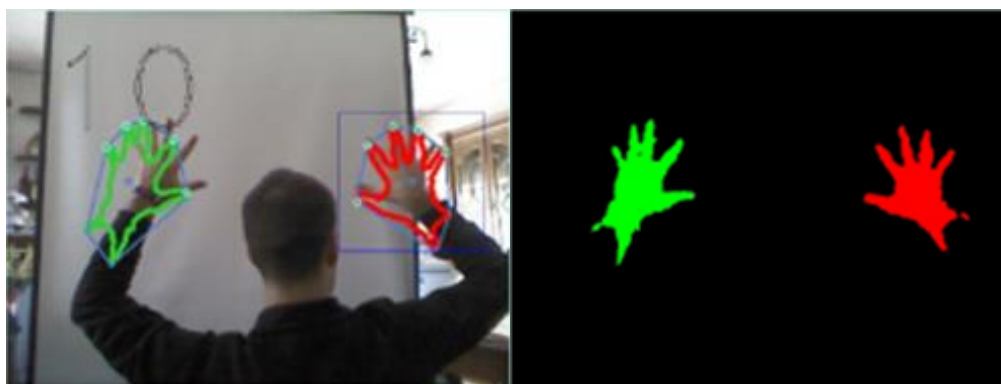


Figura 82 – Identificação de duas mãos em simultâneo.

O método utilizado para identificar qual das mãos é a direita ou a esquerda, é a sua localização em relação ao plano de trabalho. Partindo do princípio que a mão esquerda se encontra sempre à esquerda e a direita sempre à direita, é possível sabendo a localização dos dois objetos mão proceder à sua identificação.

5.7 Reconhecimento de gestos

Um gesto é uma forma do ser humano se expressar e consiste num movimento de uma parte do corpo, no espaço ao longo do tempo, e cada gesto é definido por um conjunto de características.

Para detetar um gesto é definido um processo de deteção de um determinado movimento. O princípio fundamental da captura de um gesto é a utilização das últimas coordenadas da posição da mão como entrada a aplicar num algoritmo que verifique se se trata de um potencial gesto.

A primeira classe desenvolvida para o algoritmo de deteção de gestos foi a classe *Entry* como se observa na Figura 83, que permite criar um objeto que armazena um conjunto de características como posição da mão, tempo e número de dedos.

```
using System;
using System.Windows.Shapes;

namespace Interactive_Board.Entry
{
    // Esta classe contém a posição das juntas, bem como o tempo em que são adquiridas e um elipse para a sua representação.
    public class Entry
    {
        public DateTime Time { get; set; } // Tempo de aquisição da junta
        public Vector3 Position { get; set; } // Posição das juntas
        public Ellipse DisplayEllipse { get; set; } //Representa uma elipse
        public int fingerN { get; set; } // Número de dedos
    }
}
```

Figura 83 – Classe *Entry*.

Após concluída a classe *Entry* foi desenvolvida uma classe com o nome *GestureDetector*. Esta classe contém uma lista de entradas capturadas, uma propriedade que define um atraso mínimo entre dois gestos e um evento que assinala o gesto detetado. A classe *GestureDetector* realiza a operação de descodificação e deteção dos gestos.

Como referido anteriormente, um gesto é detetado quando um conjunto de características definidas se verifica. Neste trabalho todos os gestos se caracterizam por movimentos lineares. Para as características desses movimentos lineares foi criada uma classe chamada *SwipeGestureDetector*.

Os parâmetros caracterizadores dos gestos estão descritos na Figura 84, o *SwipeMinimallLength* e o *SwipeMaximalHeight* corresponde ao comprimento que os pontos dos gestos têm que respeitar no eixo dos X e Y entre si, o *SwipeMinimalDuration* e *SwipeMaximalDuration* é o intervalo de tempo que o gesto demora a ser executado e por fim o *fingerN* o número de dedos.

```
public class SwipeGestureDetector : GestureDetector
{
    public float SwipeMinimallLength {get;set;}
    public float SwipeMaximalHeight {get;set;}
    public int SwipeMininalDuration {get;set;}
    public int SwipeMaximalDuration {get;set;}
    public int fingerN { get; set; }

    public SwipeGestureDetector(int windowSize = 10) : base(windowSize) //window Size = Número de pontos do gesto
    {
        SwipeMinimallLength = 5f;
        SwipeMaximalHeight = 3f;
        SwipeMininalDuration = 1000;
        SwipeMaximalDuration = 2000;
        fingerN = 5;
    }
}
```

Figura 84 – Classe *SwipeGestureDetector*.

Na Figura 85 encontram-se ilustrados, como exemplo, os métodos para detetar os gestos de avanço e recuo. Após a aquisição de um conjunto de entradas, numa janela de tamanho igual a 10, irão ser comparadas as características que cada uma dessas entradas têm entre si como por exemplo, o deslocamento do posicionamento das entradas nos eixos X e Y, o sentido de progressão, comprimento do gesto e o tempo de duração do gesto.

```
protected override void LookForGesture()
{
    if (ScanPositions((p1, p2) => Math.Abs(p2.Y - p1.Y) < SwipeMaximalHeight, // Altura entre pontos no eixo Y
        (p1, p2) => p2.X - p1.X > -0.01f, // Sentido de Progressão Negativo eixo X
        (p1, p2) => Math.Abs(p2.X - p1.X) > SwipeMinimallLength, // Comprimento dentre pontos final e inicial Gesto
        SwipeMininalDuration, SwipeMaximalDuration)) // Duração Mínimo e Máximo do Gesto
    {
        RaiseGestureDetected("Recuar");
        return;
    }

    if (ScanPositions((p1, p2) => Math.Abs(p2.Y - p1.Y) < SwipeMaximalHeight, // Altura entre pontos no eixo Y
        (p1, p2) => p2.X - p1.X < 0.01f, // Sentido de Progressão Positivo eixo X
        (p1, p2) => Math.Abs(p2.X - p1.X) > SwipeMinimallLength, // Comprimento dentre pontos final e inicial Gesto
        SwipeMininalDuration, SwipeMaximalDuration)) // Duração Mínimo e Máximo do Gesto
    {
        RaiseGestureDetected("Avançar");
        return;
    }
}
```

Figura 85 – Método *LookForGesture*.

A avaliação destas características num determinado conjunto de pontos irá permitir avaliar se um determinado gesto foi detetado com sucesso.

No caso do movimento de avanço é necessário que seja realizado um deslocamento da mão de um ponto inicial, como é apresentado na Figura 86, para outro ponto final apresentado na Figura 87.



Figura 86 – Ponto inicial movimento de avanço.

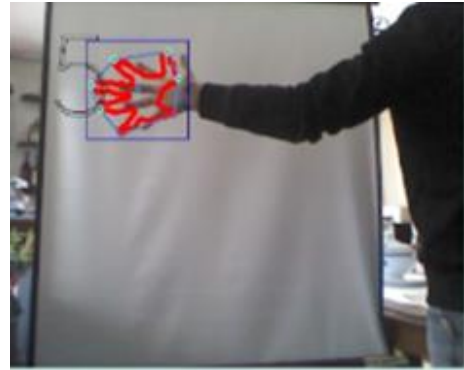


Figura 87 – Ponto final movimento de avanço.

Para que este gesto seja válido, como se pode verificar no método *LookForGesture*, os últimos 10 pontos têm que respeitar algumas características que caracterizam o gesto de avanço:

- Progressão no sentido negativo referente ao eixo X.
- Os pontos têm que se encontrar a uma distância mínima e máxima, relativamente ao eixo do Y entre si.
- Comprimento mínimo do gesto.
- Valor da duração de execução de um gesto dentro de um intervalo de tempo mínimo e máximo.

Na Figura 88 encontra-se uma imagem para uma melhor compreensão.

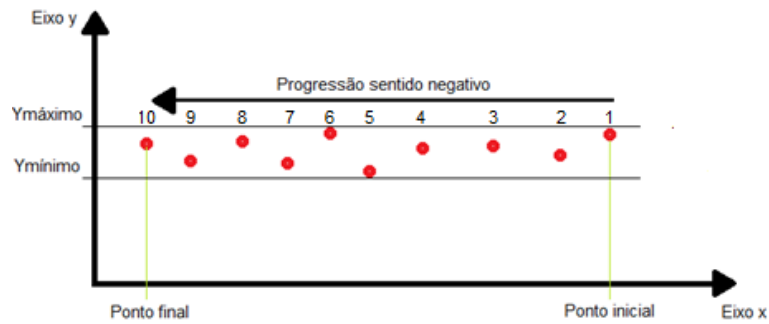


Figura 88 – Esquema de validação gesto de avanço.

Parâmetros a respeitar:

- Variação mínima no eixo x: 5.
- Variação máxima no eixo y: 3.
- Progressão: Negativa eixo x.
- Tempo mínimo entre pontos: 1000ms.
- Tempo máximo entre pontos: 2000ms.
- Número de dedos: 5.

Na Tabela 9 encontra-se o conjunto de valores registados para um exemplo de um movimento de avanço.

Tabela 9 – Percurso de posições sucessivas do gesto de avanço.

Número Ponto	Posição X/Y (pixel)	Distância em X (pixel)	Distância em Y (pixel)	Tempo (milissegundo)	Número de dedos
1	(523, 234)	-	-	-	5
2	(497, 232)	-26	-2	1200	5
3	(467, 233)	-30	+1	1400	5
4	(436, 232)	-31	-1	1500	5
5	(405, 230)	-31	-2	1500	5
6	(378, 233)	-27	+3	1400	5
7	(351, 231)	-27	-2	1400	5
8	(320, 232)	-31	+1	1500	5
9	(287, 231)	-33	-1	1600	5
10	(250, 212)	-37	+1	1700	5

5.8 Atuação

Antes de iniciar a atuação dos gestos é necessário verificar se o *Microsoft PowerPoint* se encontra a funcionar. Na Figura 89 está a inicialização e leitura do ficheiro *PowerPoint*.

```
try
{
    // Obter o objecto da aplicação PowerPoint
    pptApplication = Marshal.GetActiveObject("PowerPoint.Application") as Ppt.Application;

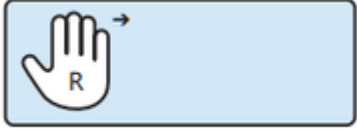

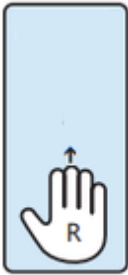
    // Se está a correr, permitir a atuação das funções
    this.First.Enabled = true;
    this.Next.Enabled = true;
    this.Prev.Enabled = true;
    this.Last.Enabled = true;
}
catch
{
    MessageBox.Show("Por favor ligar a aplicação PowerPoint primeiro", "Erro", MessageBoxButtons.OKCancel, MessageBoxIcon.Error);
}
if (pptApplication != null)
{
    // Obter o objeto da apresentação
    presentation = pptApplication.ActivePresentation;
    // Obter o conjunto de slides
    slides = presentation.Slides;
    // Obter o numero de slides
    slidescount = slides.Count;

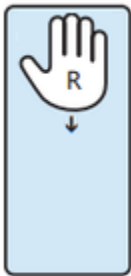
    // Obter o slide atual
    try
    {
        // Obter o objeto do slide selecionado em vista normal
        slide = slides[pptApplication.ActiveWindow.Selection.SlideRange.SlideNumber];
    }
    catch
    {
        // Obter o objeto do slide selecionado em modo de leitura
        slide = pptApplication.SlideShowWindows[1].View.Slide;
    }
}
```

Figura 89 – Inicialização do ficheiro Powerpoint.

Após ser detetado um gesto, dá-se origem a um evento para permitir realizar a correta atuação no programa. Na Tabela 10 pode-se ver a implementação do código que permite atuar no *software PowerPoint*.

Tabela 10 – Código de atuação de gestos no sistema.

 <p>Recuo</p>	<p>Ação de <i>slide</i> anterior</p> <pre>// Ação slide Anterior if(Gesture=="Recuo"){ slideIndex = slide.SlideIndex - 1; if (slideIndex >= 1) { try { slide = slides[slideIndex]; slides[slideIndex].Select(); } catch { pptApplication.SlideShowWindows[1].View.Previous(); slide = pptApplication.SlideShowWindows[1].View.Slide; } } } else { MessageBox.Show("It is already First Page"); } }</pre>
 <p>Avanço</p>	<p>Ação de <i>slide</i> seguinte</p> <pre>// Ação slide Seguinte if(Gesture=="Avanço"){ slideIndex = slide.SlideIndex + 1; if (slideIndex > slidescount) { MessageBox.Show("It is already last page"); } else { try { slide = slides[slideIndex]; slides[slideIndex].Select(); } catch { pptApplication.SlideShowWindows[1].View.Next(); slide = pptApplication.SlideShowWindows[1].View.Slide; } } } }</pre>
 <p>Cima</p>	<p>Ação de <i>slide</i> inicial.</p> <pre>// Ação slide inicial if(Gesture=="Cima"){ try { // Chama o método Select() para selecionar o primeiro slide em vista normal slides[1].Select(); slide = slides[1]; } catch { // Transforma a primeira página em vista de leitura pptApplication.SlideShowWindows[1].View.First(); slide = pptApplication.SlideShowWindows[1].View.Slide; } } }</pre>

 <p>Baixo</p>	<p>Ação de último <i>slide</i>.</p> <pre>// Ação slide Final if(Gesture=="Baixo"){ try { // Chama o método Select() para selecionar o ultimo slide em vista normal slides[slidescount].Select(); slide = slides[slidescount]; } catch { // Transforma a ultima página em vista de leitura pptApplication.SlideShowWindows[1].View.Last(); slide = pptApplication.SlideShowWindows[1].View.Slide; } }</pre>
--	--

5.9 Desenho e escrita interativa

Uma funcionalidade desenvolvida é a possibilidade de desenho e escrita interativa. Foi necessário inicialmente desenvolver uma matriz para que fosse possível guardar a posição dos pontos percorridos com o objetivo de serem utilizados, tratados e acrescentados na imagem.

Sendo a aquisição destes pontos bastante precisa em tempo real, surgem algumas discontinuidades nas linhas de desenho como seria de esperar, sendo por isso necessário realizar uma operação de tratamento dos pontos com o objetivo de efetuar a conexão entre os mesmos e corrigir as discontinuidades.

Esta operação de tratamento recorre à utilização de operações matemáticas simples para que sejam traçadas linhas entre os pontos e expandir o tamanho do ponto, possibilitando a obtenção de uma melhor perceção pelo utilizador, na sua projeção.

5.10 Comando por voz

Com a ajuda do *Microsoft Speech API* foi concebido um sistema de comando de voz para realizar algumas operações que auxiliem o utilizador na interatividade com o computador. Para isso desenvolveu-se uma classe chamada *VoiceCommander* que é construída em torno da informação proveniente da captura de som do sensor Kinect

Esta informação é compatível com o *Microsoft.Speech* instalado com o SKD para *Windows* e o seu fluxo de informação pode ser utilizado pela classe *SpeechRecognitionEngine* para detetar palavras.

A utilização desta classe é bastante simples, apenas é necessário inicializá-la. Observe-se a Figura 90, com as palavras que se pretende detetar.

```
voiceCommander = new VoiceCommander("start", "stop", "close", "Forward", "Back", "Up", "Down", "In", "Out", "Green", "Blue", "Clear" );
```

Figura 90 – Inicialização VoiceCommaner.

Na Figura 91 é inicializada a captura do som através de um método *Start* na classe *VoiceCommander*.

```
//----- Método para inicializar a captura das palavras por som-----  
//-----  
void StartVoiceCommander()  
{  
    var kinectSensor = KinectSensor.KinectSensors[0]; // Cria uma variavel do tipo KinectSensor ao primeiro Sensor Kinect Detectado  
    voiceCommander.Start(kinectSensor); // Inicia o método Start da classe Voice Commander para dar início ao comando por voz  
}
```

Figura 91 – Inicialização da captura do som.

De seguida é necessário lidar com o evento, cada palavra detetada gera uma operação no interface como se pode verificar na Figura 92.



Figura 92 – Operações para as palavras detetadas.

Capítulo 6

Conclusões e trabalho futuro

6.1 Conclusão

O objetivo da dissertação era o reconhecimento de gestos que permitam realizar apresentações informáticas com recurso ao sensor Kinect, permitindo atuar em quadros interativos de uma forma intuitiva e mais simples.

A solução proposta atinge este objetivo e permite que o sistema desenvolvido atue no ambiente de um computador através do reconhecimento de gestos realizados com as mãos do utilizador.

O sensor Kinect não possibilita a deteção e rastreio das mãos, apenas a localização das juntas referentes às mesmas, logo para isso, foi desenvolvido um método que realiza essas funções e extraia as suas características.

Com este sistema é possível detetar vários tipos de gestos, correspondendo a cada gesto uma diferente atuação no sistema, alterando a forma como o utilizador interage com um quadro interativo.

Um processo importante era a deteção e rastreio das mãos, de forma a serem retiradas as suas características e permitir ao sistema reconhecer e atuar adequadamente.

Este processo deu início com um pré-processamento e segmentação das mãos pela cor da pele e profundidade. O método utilizado para a segmentação das mãos pela cor da pele, mostrou-se bastante eficiente, permitindo obter uma boa deteção como foi demonstrado. De seguida, foi efetuada uma calibração para minimizar o erro de *offset* existente entre a imagem RGB e a informação da profundidade. A extração das características da mão foi realizada recorrendo à utilização do *Emgu CV*, resultando num processo muito eficaz a partir do qual é possível obter resultados bastante precisos.

A deteção dos gestos propostos neste trabalho é efetuada com um bom grau de certeza, bem como a correspondente atuação no sistema.

Efetuando a combinação destes vários processos é possível dizer que o objetivo proposto foi atingido com sucesso.

Apesar disso, verificaram-se algumas limitações no desenvolvimento do trabalho. Quando a luminosidade na área de reconhecimento dos gestos se encontra bastante reduzida, é possível que não exista uma boa segmentação das mãos pela cor da pele, não sendo efetuado um bom reconhecimento.

Existem ainda alguns problemas intrínsecos ao sensor, como por exemplo a variação da informação relativa à profundidade se apresentar pouco precisa para um mesmo ponto da imagem, assim como o *offset* que esta informação tem relativamente à imagem RGB e o tempo de aquisição dos *streams* de vídeo e profundidade se apresentarem com algum desfasamento entre eles.

6.2 Trabalho Futuro

Apresentam-se algumas propostas de possíveis melhoramentos que podem ser realizadas futuramente e implementadas no sistema desenvolvido. Assim é importante:

- Efetuar uma sincronização entre a camera RGB e a informação da profundidade, por exemplo, a aquisição da imagem RGB e profundidade não é efetuada no mesmo instante e a receção no computador através da porta USB causa um erro de aproximadamente 16 milissegundos entre *frames*. Para este trabalho torna-se essencial que a informação seja adquirida ao mesmo tempo para diminuir possíveis erros;
- Melhorar o método pelo qual é identificada a mão esquerda e direita. Pode ser utilizado a orientação dos dedos polegares, bem como a localização do dedo polegar em relação ao indicador e com isso distinguir qual das mãos é detetada;
- Detecção da superfície do quadro interativo. Através de um detetor de cantos como o detetor de *Harris*, simultaneamente com a deteção da cor projetada. Utilizando este processo pode-se projetar uma imagem de cor definida, e através da deteção da mesma é possível realizar uma binarização da imagem adquirida e de seguida, obter a localização dos cantos da projeção, com recurso ao detetor de *Harris*;

- Desenvolver um maior número de gestos, para possibilitar uma maior gama de diferentes tipos de atuações no *Microsoft PowerPoint*, bem como aplicar a outros programas como por exemplo, *PDF Reader*, Navegadores de internet, *Microsoft Word*, *Windows Media Player*, programas de comunicação e partilha de vídeo e voz, assim como muitos outros.
- Identificação de gestos a partir de imagens padrão. A correspondência de padrões envolve a utilização de uma base de dados de gestos gravados que servem como padrões com os quais os gestos detetados podem ser comparados. Utilizando este método, os gestos são representados por imagens bidimensionais em que cada ponto é uma posição no tempo de um conjunto específico. É necessário desenvolver um algoritmo de correspondência de padrões para determinar se um padrão corresponde ao desenho atual, e se assim se verificar, então o gesto é detetado.

Referências Bibliográficas

[1] GO-Gulf – 60 Seconds: Things That Happen Every Sixty Seconds [Infographic] [on-line]. 2011. Disponível a 14-11-2014 em <<http://www.go-gulf.com/blog/60-seconds-v2/>>.

[2] Silva, Marco - Projeto de Quadro Interativo de Baixo Custo Utilizando o Comando da Wii. Porto: [s.n.], 2008. Dissertação apresentada à Faculdade de Engenharia da Universidade do Porto para a obtenção do Grau de Mestre em Engenharia Eletrotécnica e de Computadores.

[3] Techtudo – Resistive Touchscreen [on-line]. 2013. Disponível a 14-11-2014 em <<http://www.techtudo.com.br/artigos/noticia/2013/06/qual-e-diferenca-entre-tela-capacitiva-e-tela-resistiva-entenda.html>>.

[4] Strongarm – Infrared [on-line]. 2014. Disponível a 14-11-2014 em <<http://www.strongarm.com/systemtouchscreens.html>>.

[5] Smart – Sistema de lousa interativa SMART Board 600i [on-line]. 2014. Disponível a 14-11-2014 em <<https://smarttech.com/br/Solutions/Higher+Education+Solutions/Products+for+higher+education/Interactive+whiteboards+and+displays/SMART+Board+interactive+whiteboards/SMART+Board+600i>>.

[6] Tramaweb – Quadros interativos de projeção traseira da SMART Technologies são soluções de alta qualidade para ambientes sofisticados [on-line]. 2005. Disponível a 14-11-2014 em <<http://www.tramaweb.com.br/press-release/2819/quadros-interativos-de-projecao-traseira-da-smart-technologies-sao-solucoes-de-alta-qualidade-ambientes-sofisticados.aspx>>.

[7] Gadgetlux – Samsung PS51E8000 plasma TV offers landmark HD [on-line]. 2012. Disponível a 14-11-2014 em <<http://www.gadgetlux.co.uk/root/review-samsung-e8000-51-inch-plasma-tv-offers-cinema-thrills-and-more/>>.

[8] Ministério da Educação - Plano Tecnológico da Educação [on-line]. 2007. Disponível a 17-11-2013 em www.pte.gov.pt/idc/idcplg?IdcService=GET_FILE&dID=13429&dDocName=002386.

[9] Ministério da Educação - Kit Tecnológico [on-line]. 2013. Disponível a 17-11-2013 em <http://www.pte.gov.pt/pte/PT/Projectos/Projecto/index.htm?proj=6>.

[10] Infogroup, ORC International - Harnessing Technology schools survey: 2010 [on-line]. 2010. Disponível a 17-11-2013 em http://dera.ioe.ac.uk/1544/1/becta_2010_htss_report.pdf.

[11] Kean, Sean, Jonathan Hall e Phoenix Perry – Meet the Kinect: An Introduction to Programming Natural User Interfaces, Technology in Action, 2011.

[12] Microsoft Developer Network - Kinect for Windows Human Interface Guidelines v1.8.0 [on-line]. 2010. Disponível a 17-11-2013 em <http://msdn.microsoft.com/en-us/library/jj663791.aspx>.

[13] David Catuhe - Programming with the Kinect™ for Windows® Software Development Kit: Add gesture and posture recognition to your applications, O'Reilly Media, Inc., 2012.

[14] Bayer, Bryce E. - Color imaging array, U.S. Patent No. 3,971,065., 20 de Julho de 1976.

[15] Microsoft BizSpark - The microsoft accelerator for Kinect [on-line]. 2010. Disponível a 24-02-2014 em <http://www.microsoft.com/bizspark/kinectaccelerator/>.

[16] Jintronix - Rehabilitation System [on-line]. 2012. Disponível a 24-02-2014 em <http://www.jintronix.com/>.

[17] ManCTL – Skanect: 3D Scanning [on-line]. 2011. Disponível a 24-02-2014 em <http://skanect.manctl.com/>.

- [18] Ikkos – Ikkos Training [on-line]. 2013. Disponível a 24-02-2014 em <<http://ikkos.com/how-ikkos-works/>>.
- [19] Ingematica SA – Kimetric [on-line]. 2012. Disponível a 24-02-2014 em <<http://www.kimetric.com/staff/whatis/>>.
- [20] NConnex – Picksureit [on-line]. 2012. Disponível a 24-02-2014 em <<http://nconnex.com/>>.
- [21] Styku – SmartFit [on-line]. 2013. Disponível a 25-02-2014 em <<http://www.styku.com/business/>>.
- [22] Ubi. – Ubi Interactive [on-line]. 2013. Disponível a 10-03-2014 em <<http://www.ubi-interactive.com/>>.
- [23] Atlas5D – Atlas5D [on-line]. 2013. Disponível a 10-03-2014 em <<http://www.atlas5d.com/>>.
- [24] Gestsure – Gestsure [on-line]. 2014. Disponível a 10-03-2014 em <<http://www.gestsure.com/>>.
- [25] MIT – Kinect Hand Detection [on-line]. 2010. Disponível a 10-03-2014 em <<http://video.mit.edu/watch/kinect-hand-detection-12073/>>.
- [26] Martinkauppi, J. Birgitta, Maricor N. Soriano e Mika V. Laaksonen - Behavior of skin color under varying illumination seen by different cameras at different color spaces, Photonics West 2001 Electronic Imaging. International Society for Optics and Photonics, 2001.
- [27] Zarit, Benjamin D., Boaz J. Super, and Francis KH Quek - Comparison of five color models in skin pixel classification, Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 1999. Proceedings. International Workshop on. IEEE, 1999.

[28] Lopes, João Manuel Brisson - Cor e Luz, 2000.

[29] Betaface – Face Recognition Software [on-line]. 2007. Disponível a 17-03-2014 em <<http://betaface.com/>>.

[30] Tang, Matthew – Recognizing hand gestures with microsoft's Kinect, Palo Alto: Department of Electrical Engineering of Stanford University:[sn], 2011.

[31] Jones, Michael J., and James M. Rehg. - Statistical color models with application to skin detection, International Journal of Computer Vision 46.1, 2002.

[32] Phung, Son Lam, Abdesselam Bouzerdoun, and D. Chai Sr. - Skin segmentation using color pixel classification: analysis and comparison, Pattern Analysis and Machine Intelligence, IEEE, 27 de Janeiro de 2005.

[33] Shin, Min C., Kyong I. Chang, and Leonid V. Tsap. - Does colorspace transformation make any difference on skin detection?, Applications of Computer Vision, 2002.

[34] Aly A Farag. - Skin-based Face Detection [on-line]. 2009. Disponível a 19-03-2014 em <<http://www.cvip.uofl.edu/wwwcvip/education/ECE523/Spring%202011/Lec4%20-%20Skin%20Detection.pdf>>.

[35] Conaire, Ciarán O., Noel E. O'Connor, and Alan F. Smeaton.- Detector adaptation by maximising agreement between independent data sources, Computer Vision and Pattern Recognition, 2007.

[36] Montalvão, Jugurta, Lucas Molina, and Jânio Canuto.- Robust hand image processing for biometric application, Pattern Analysis and Applications 13.4, 2010.

[37] Khoshelham, Kourosh. - Accuracy analysis of kinect depth data, ISPRS workshop laser scanning. Vol. 38. No. 5. 2011.

[38] MathWorks – Multidimensional Arrays [on-line]. 2014. Disponível a 24-02-2014 em <<http://www.mathworks.com/help/matlab/math/multidimensional-arrays.html>>.

Apêndice 1

Ensaaios HSV, YUV e Método Probabilístico

A.1 Ensaaios Mãos

A.1.1 Mão Número 1

A.1.1.1 Método HSV

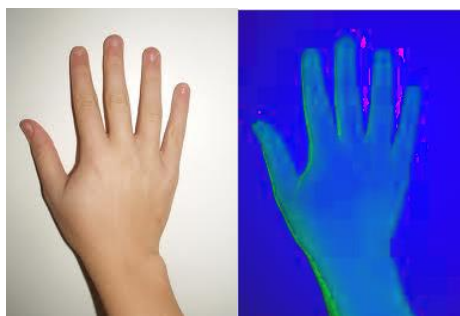


Figura 93 – Transformação RGB para HSV da mão número 1.

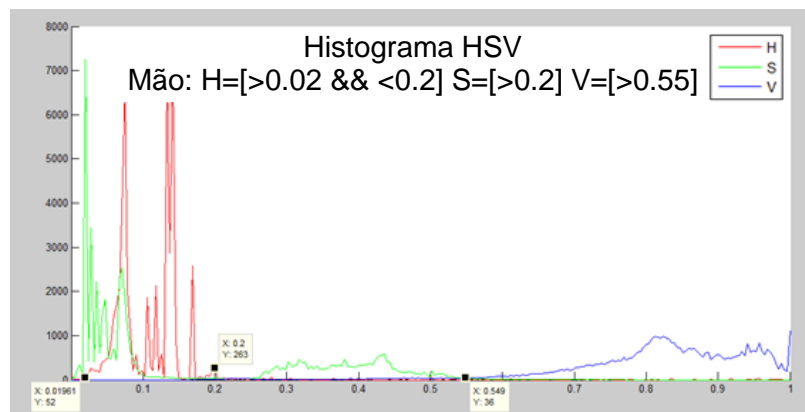


Figura 94 – Histograma HSV da imagem da mão número 1.



Figura 95 – Segmentação HSV da mão número 1.

A.1.1.2 Método YCbCr

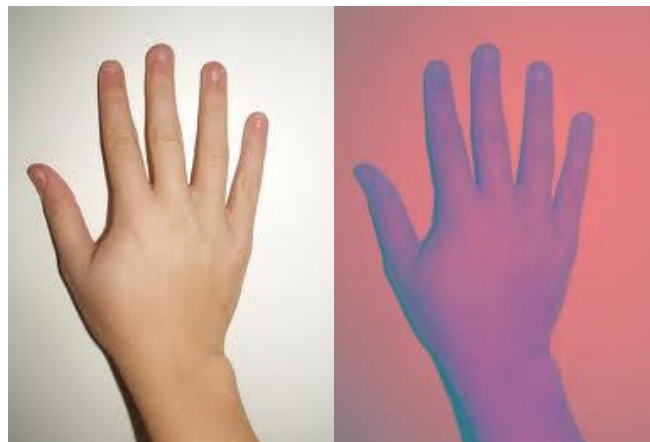


Figura 96 – Transformação RGB para YCbCr da mão número 1.

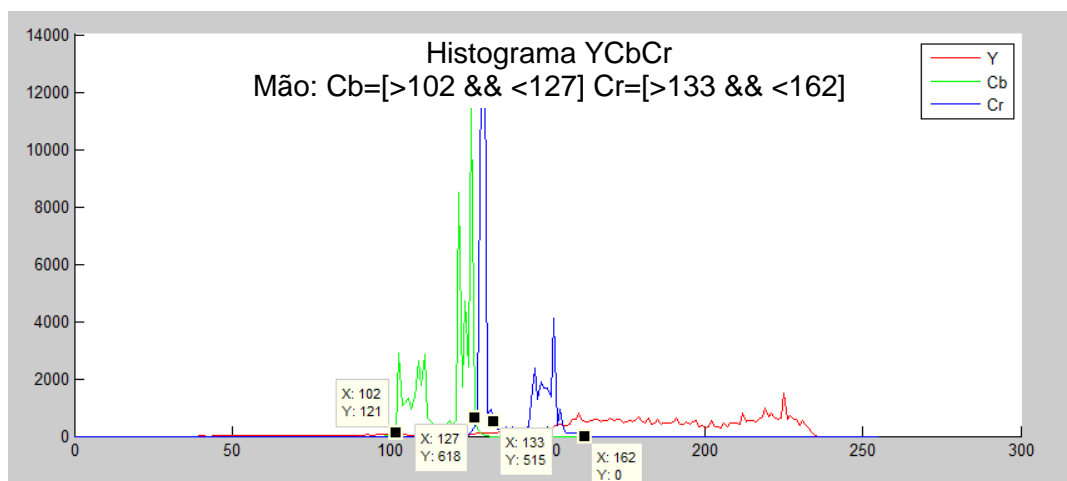


Figura 97 – Histograma YCbCr da mão número 1.



Figura 98 – Segmentação YCbCr da mão número 1.

A.1.1.3 Método Probabilístico

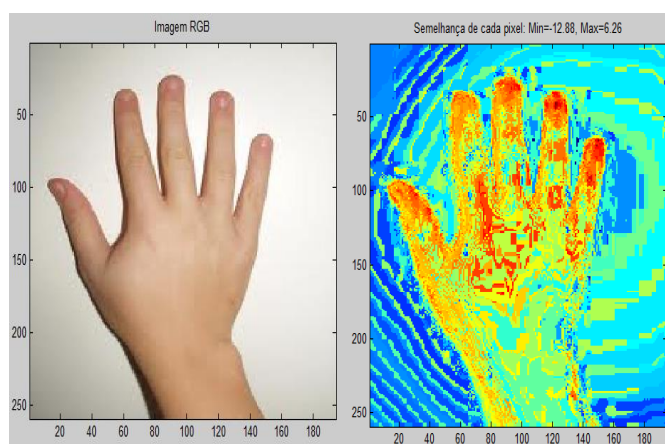


Figura 99 – Transformação de RGB para método probabilístico da mão número 1.

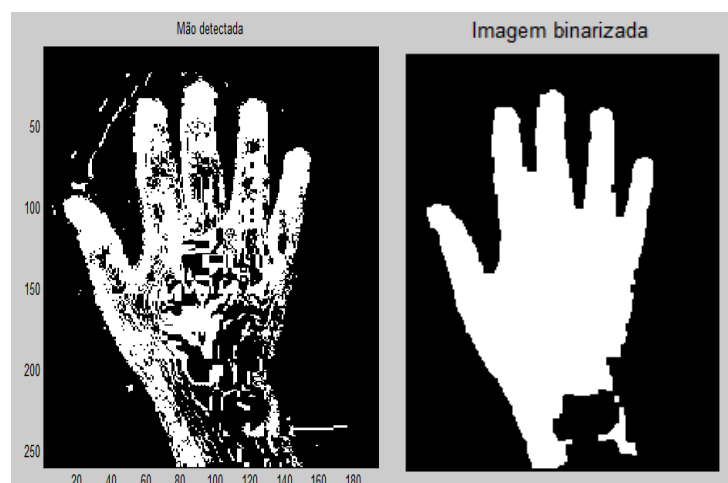


Figura 100 – Resultado das operações morfológicas na imagem binarizada da mão número 1.



Figura 101 – Imagem da mão número 1 obtida após método probabilístico.

A.1.2 Mão Número 2

A.1.2.1 Método HSV

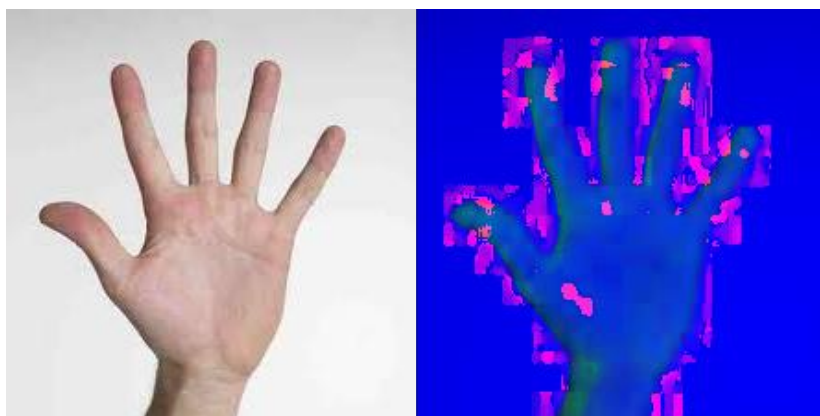


Figura 102 – Transformação RGB para HSV da mão número 2.

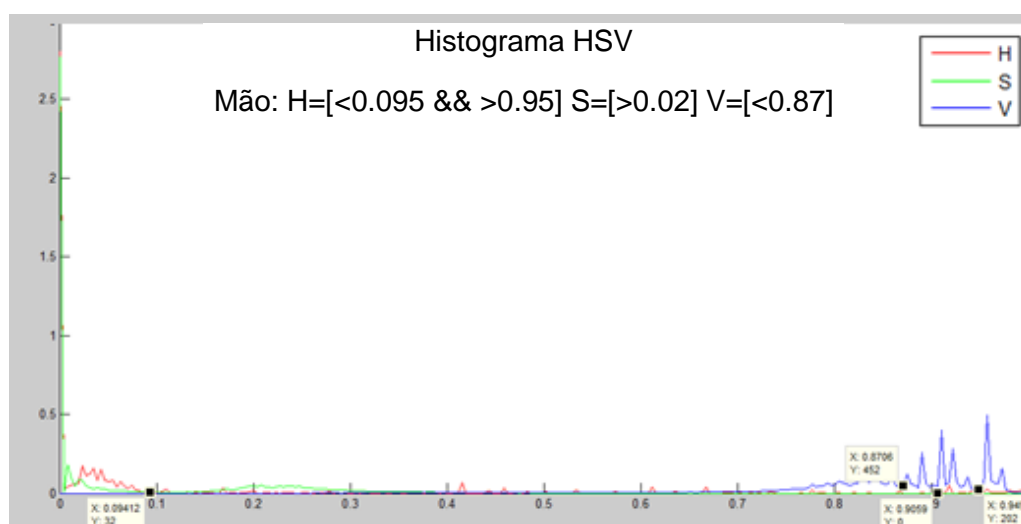


Figura 103 – Histograma HSV da mão número 2.



Figura 104 – Segmentação HSV da mão número 2.

A.1.2.2 Método YCbCr

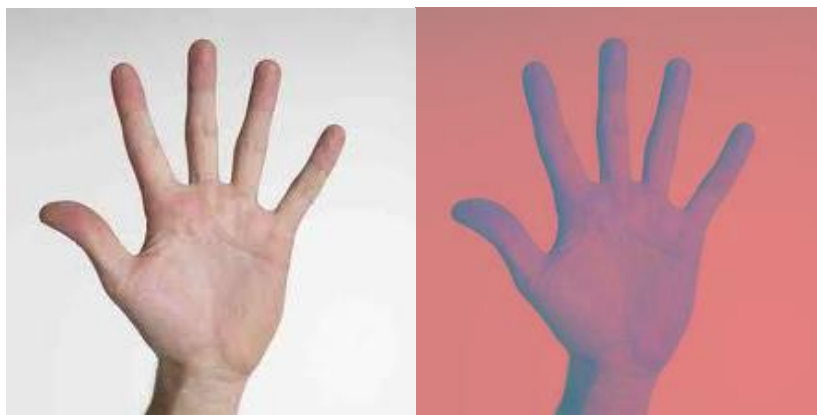


Figura 105 – Transformação RGB para YCbCr da mão número 2.

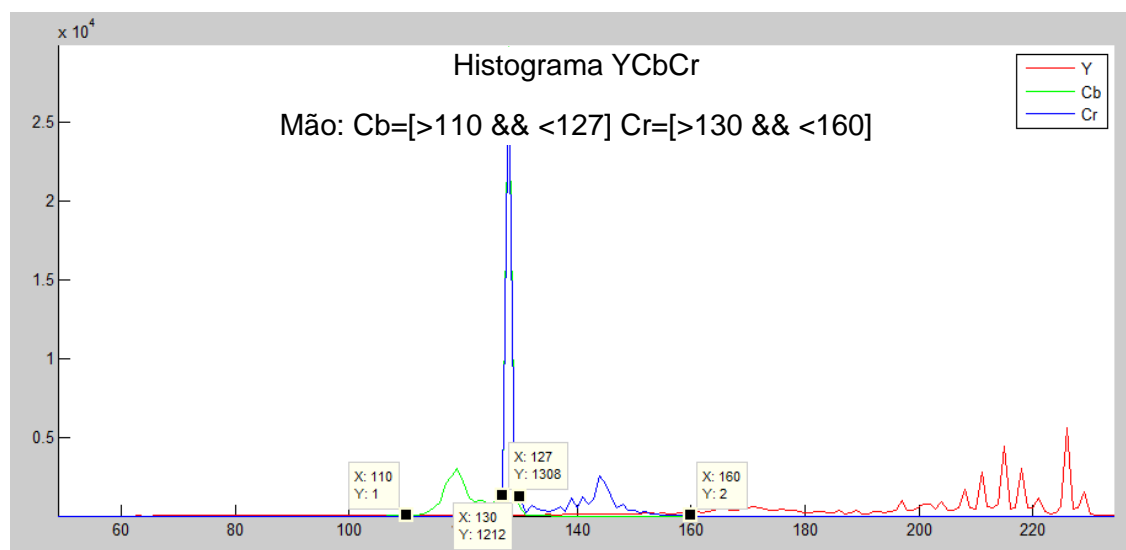


Figura 106 – Histograma YCbCr da mão número 2.



Figura 107 – Segmentação YCbCr da mão número 2.

A.1.2.3 Método Probabilístico

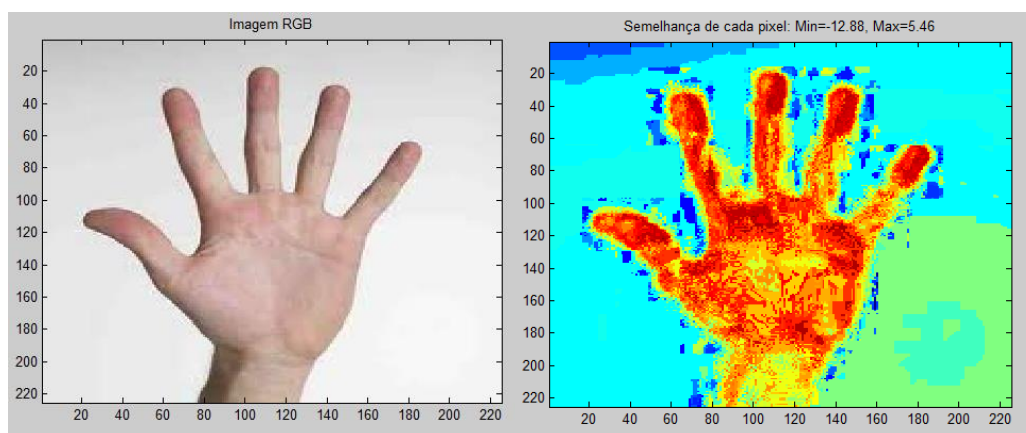


Figura 108 – Transformação de RGB para método probabilístico da mão número 2.

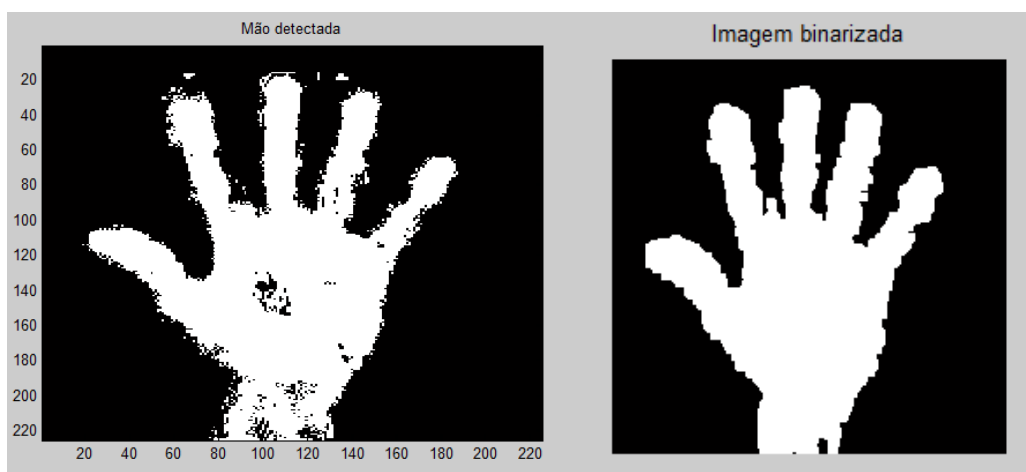


Figura 109 – Resultado das operações morfológicas na imagem binarizada da mão número 2.



Figura 110 – Imagem da mão número 2 obtida após método probabilístico.

A.1.3 Mão Número 3

A.1.3.1 Método HSV

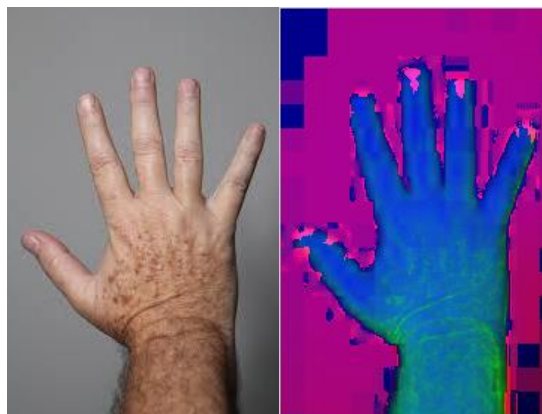


Figura 111 – Transformação RGB para HSV da mão número 3.

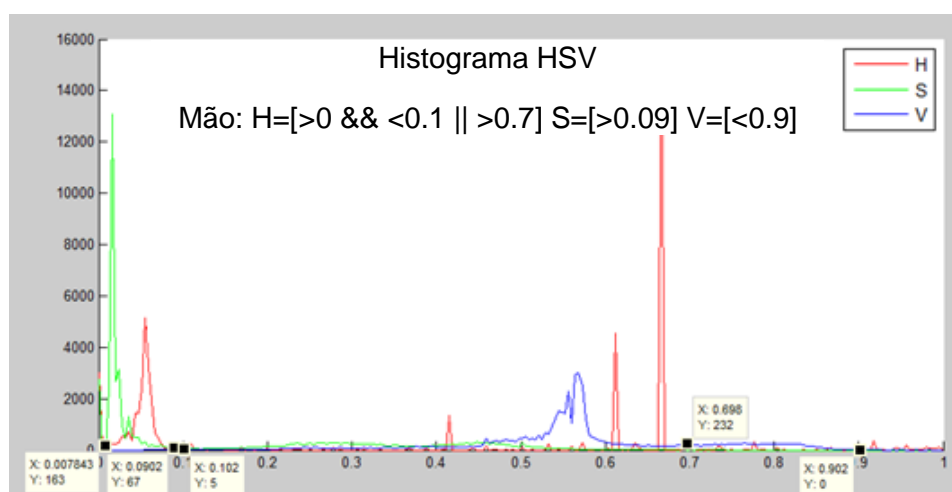


Figura 112 – Histograma HSV da mão número 3.



Figura 113 – Segmentação HSV da mão número 3.

A.1.3.2 Método YcbCr



Figura 114 – Transformação RGB para YCbCr da mão número 3.

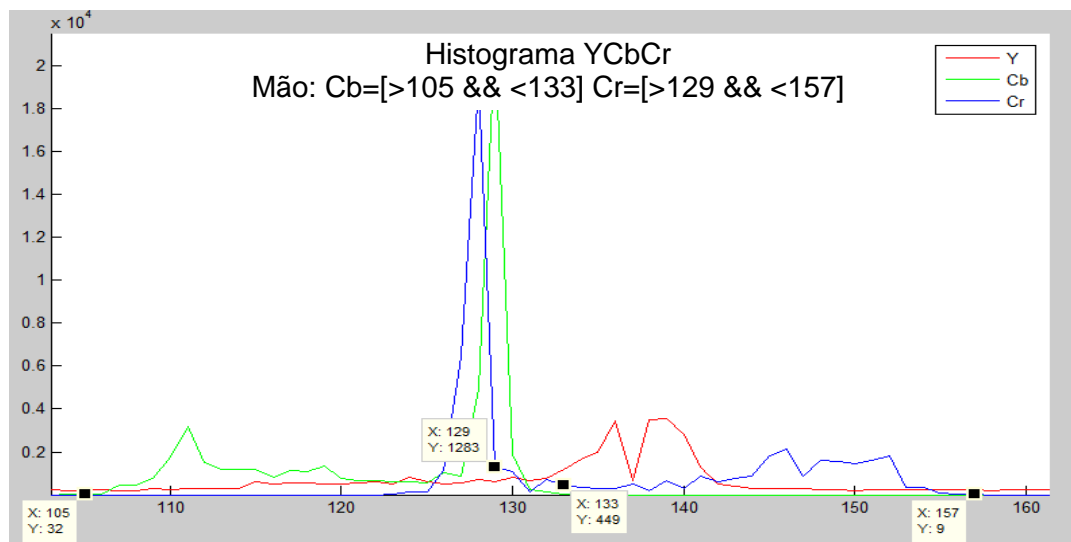


Figura 115 – Histograma YCbCr da mão número 3.



Figura 116 – Segmentação YCbCr da mão número 3.

A.1.3.3 Método Probabilístico

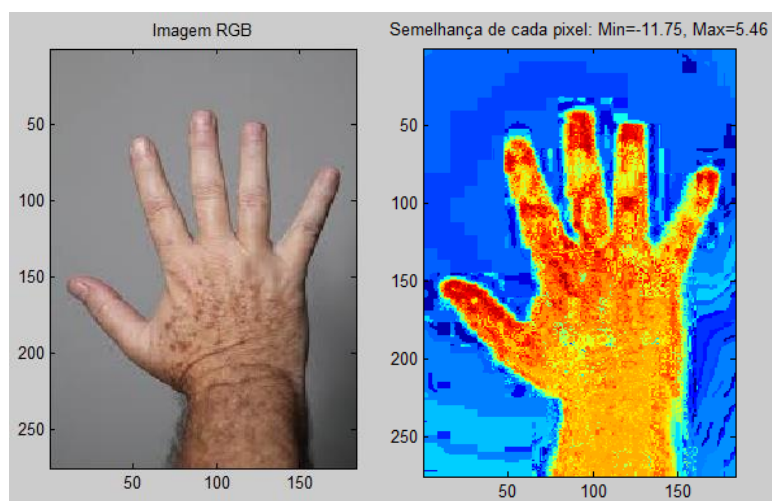


Figura 117 – Transformação de RGB para método probabilístico da mão número 3.

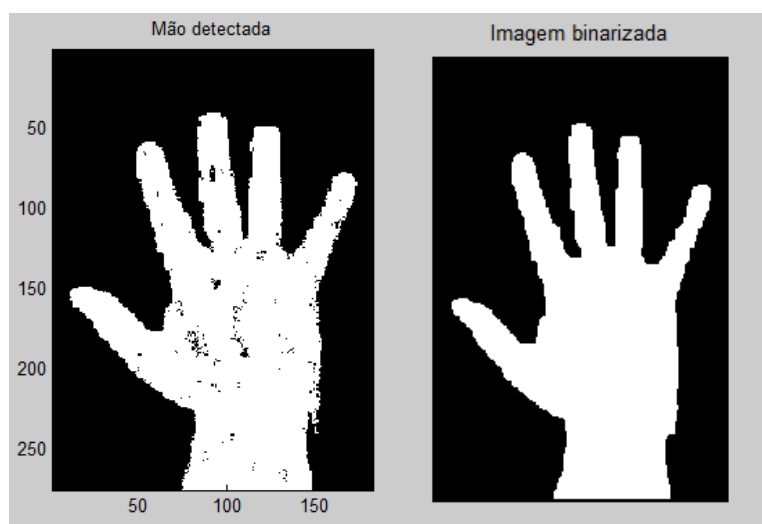


Figura 118 – Resultado das operações morfológicas na imagem binarizada da mão número 3.



Figura 119 – Imagem da mão número 3 obtida após método probabilístico.

A.1.4 Mão Número 4

A.1.4.1 Método HSV

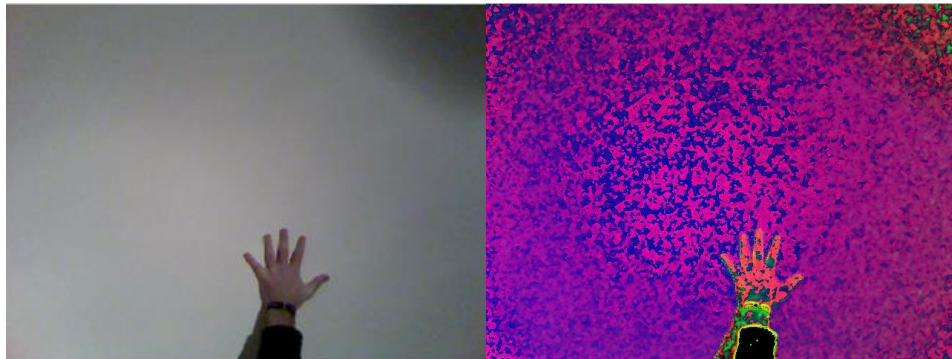


Figura 120 – Transformação RGB para HSV da mão número 4.

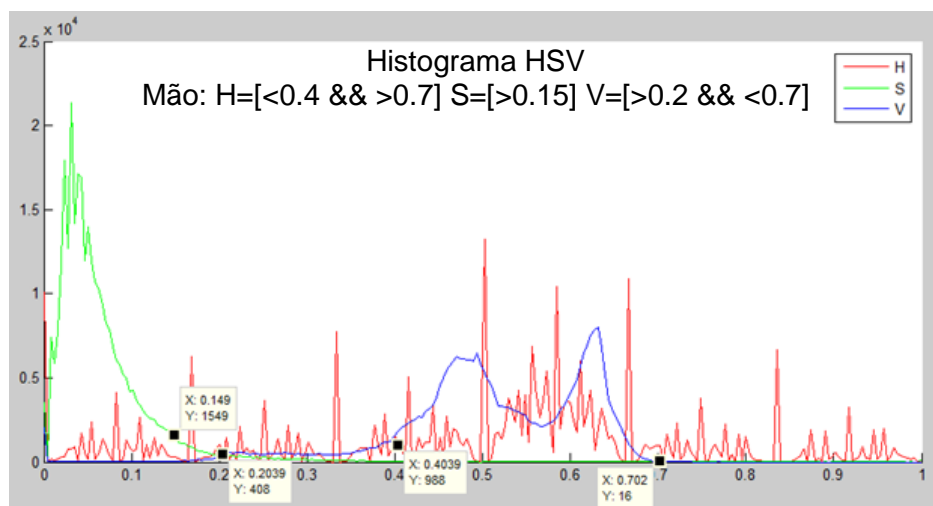


Figura 121 – Histograma HSV da mão número 4.



Figura 122 – Segmentação HSV da mão número 4.

A.1.4.2 Método YCbCr

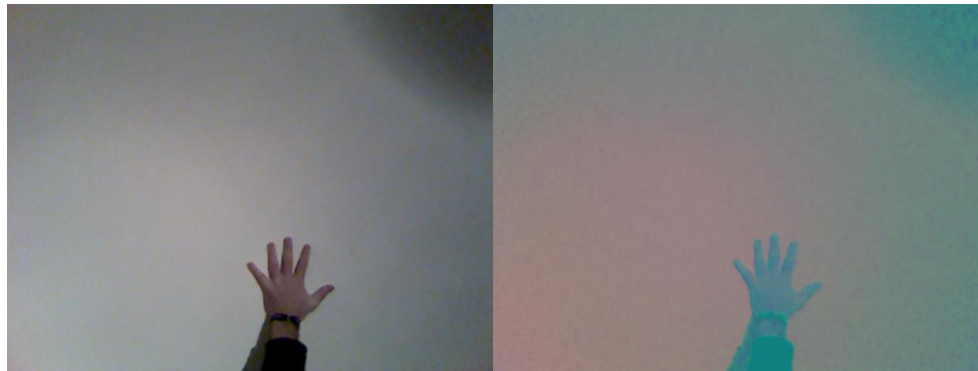


Figura 123 – Transformação RGB para YCbCr da mão número 4.

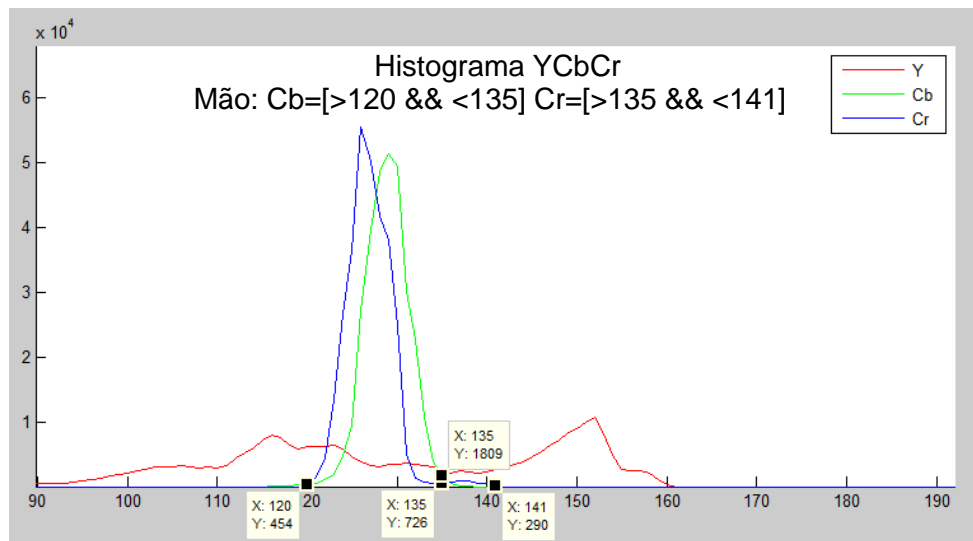


Figura 124 – Histograma YCbCr da mão número 4.



Figura 125 – Segmentação YCbCr da mão número 4.

A.1.4.3 Método Probabilístico

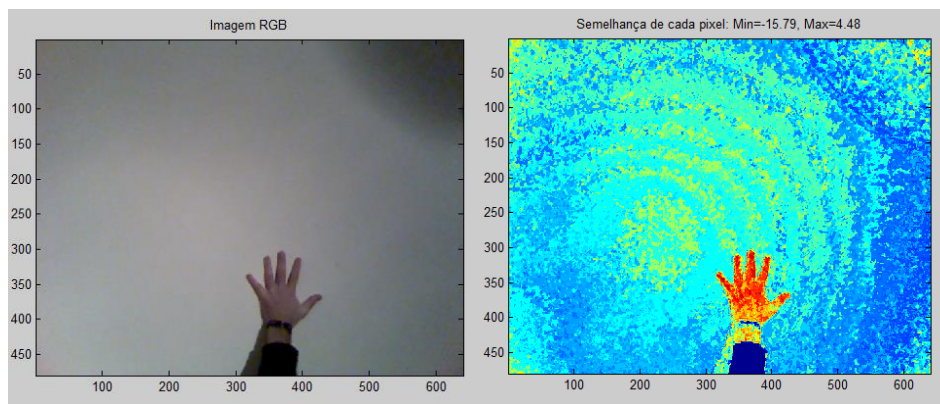


Figura 126 – Transformação de RGB para método probabilístico da mão número 4.

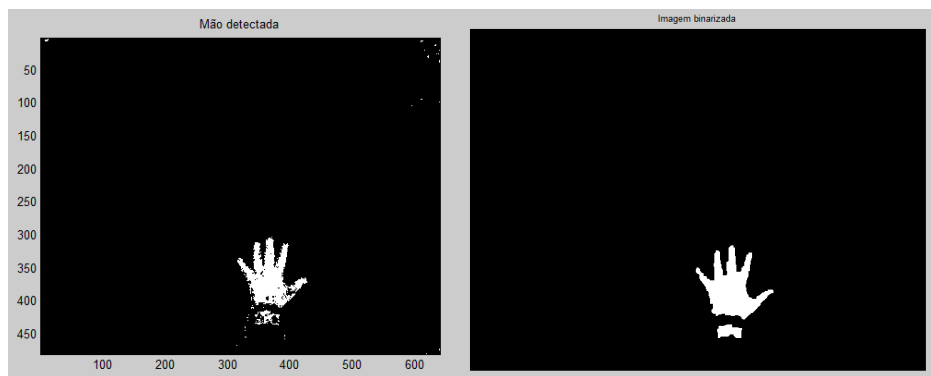


Figura 127 – Resultado das operações morfológicas na imagem binarizada da mão número 4.

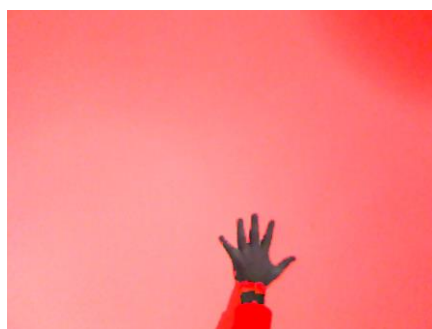


Figura 128 – Imagem da mão número 4 obtida após método probabilístico.

A.1.5 Mão Número 5

A.1.5.1 Método HSV

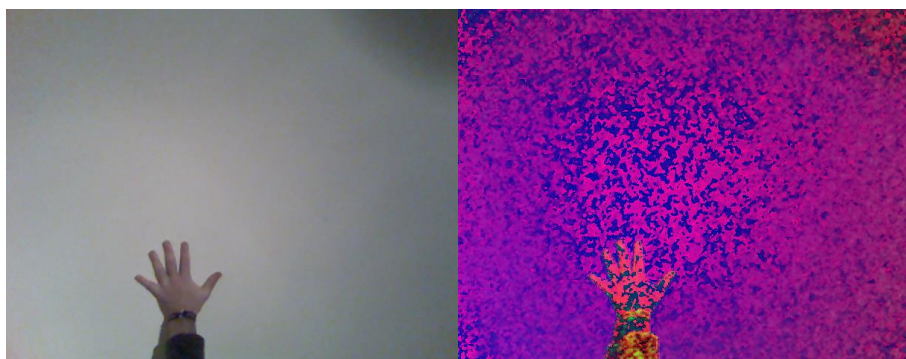


Figura 129 – Transformação RGB para HSV da mão número 5.

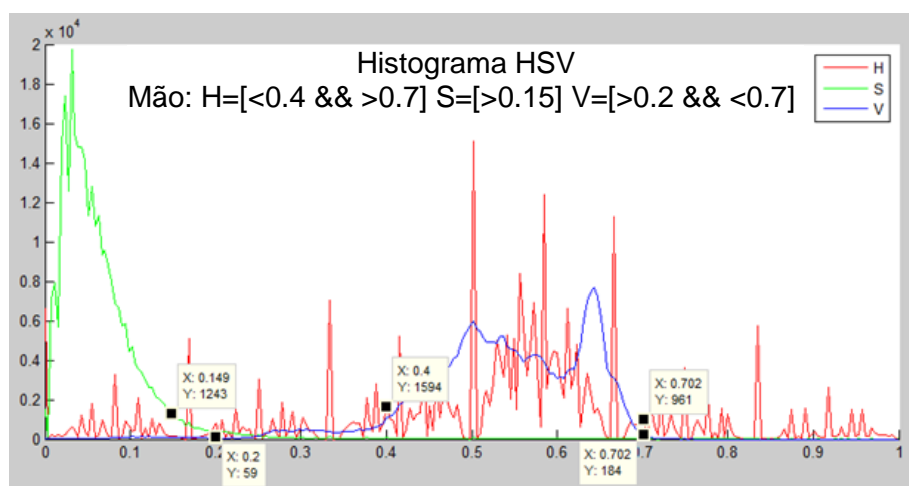


Figura 130 – Histograma HSV da mão número 5.

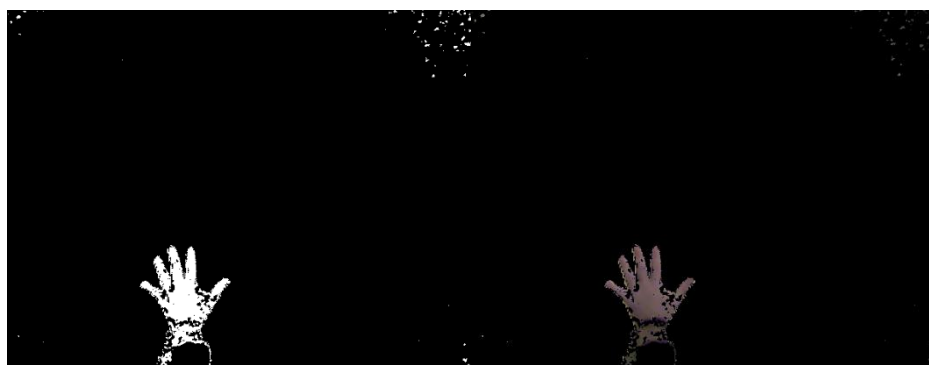


Figura 131 – Segmentação HSV da mão número 5.

A.1.5.2 Método YCbCr

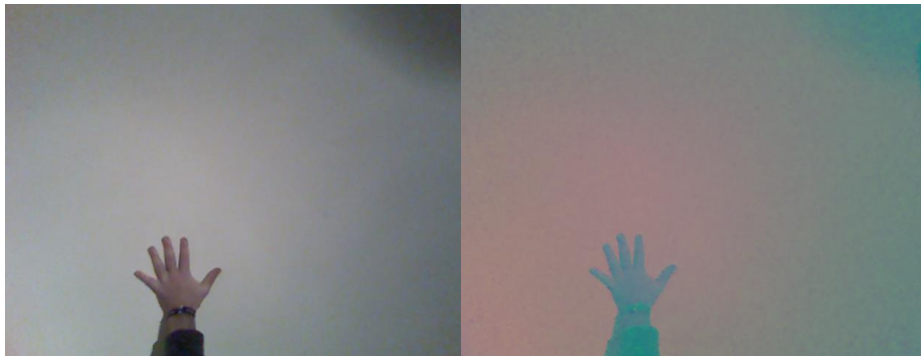


Figura 132 – Transformação RGB para YCbCr da mão número 5.

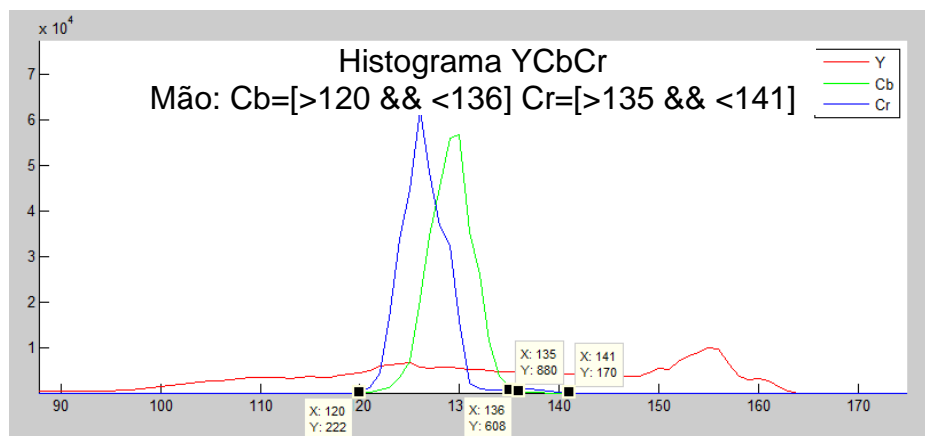


Figura 133 – Histograma YCbCr da mão número 5.



Figura 134 – Segmentação YCbCr da mão número 5.

A.1.5.3 Método Probabilístico

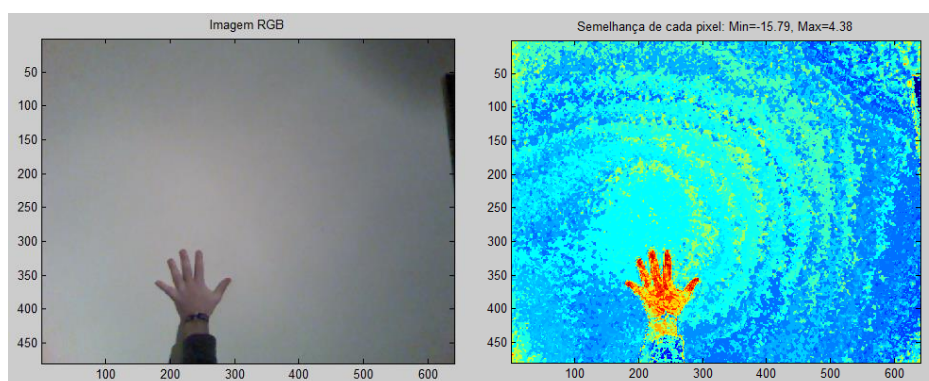


Figura 135 – Transformação de RGB para método probabilístico da mão número 5.

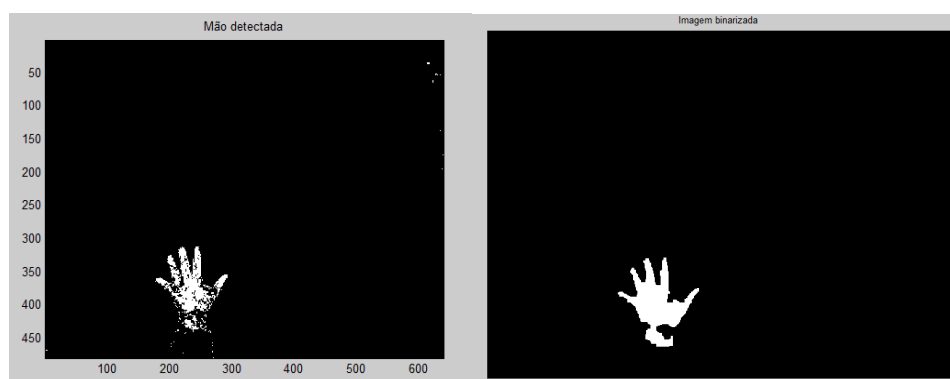


Figura 136 – Resultado das operações morfológicas na imagem binarizada da mão número 5.

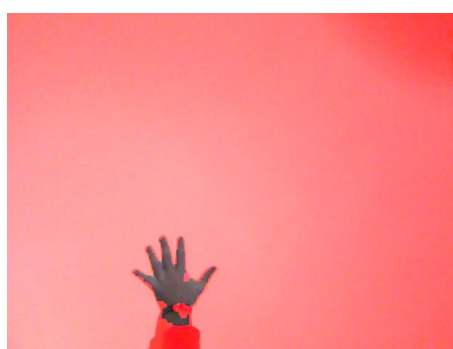


Figura 137 – Imagem da mão número 5 obtida após método probabilístico.

A.1.6 Mão Número 6

A.1.6.1 Método HSV

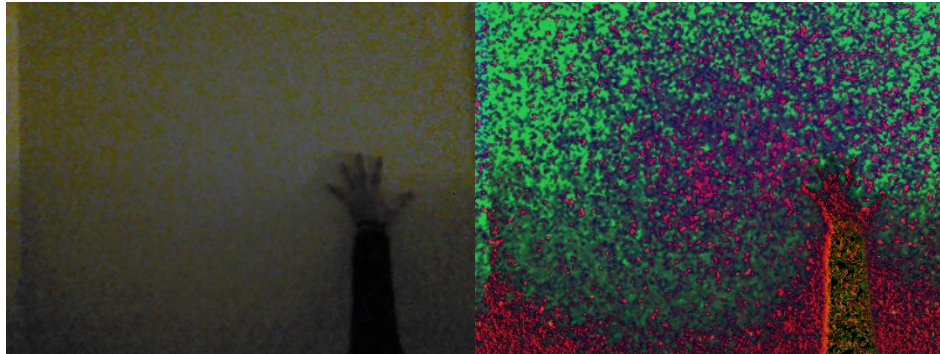


Figura 138 – Transformação RGB para HSV da mão número 6.

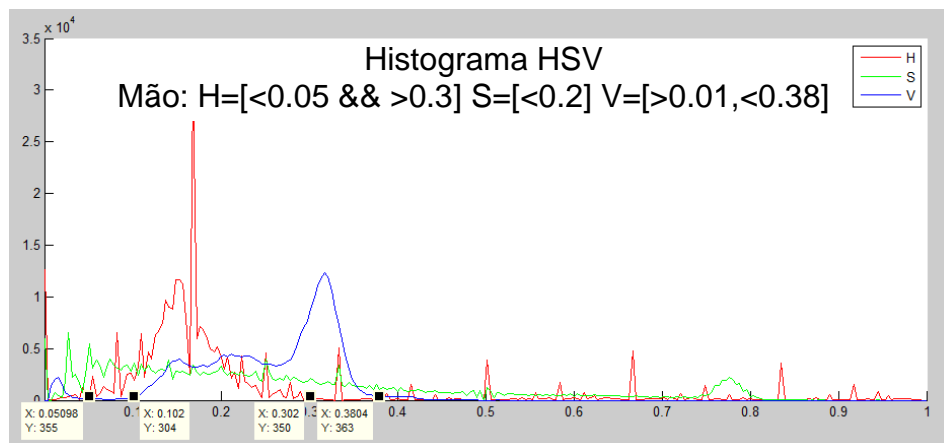


Figura 139 – Histograma HSV da mão número 6.

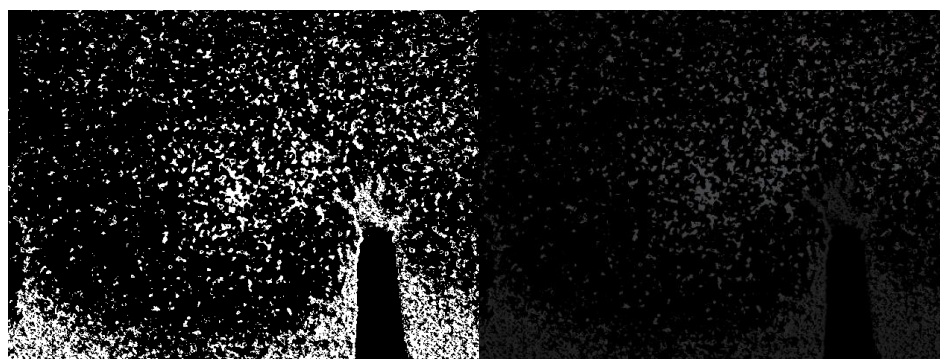


Figura 140 – Segmentação HSV da mão número 6.

A.1.6.2 Método YCbCr



Figura 141 – Transformação RGB para HSV da mão número 6.

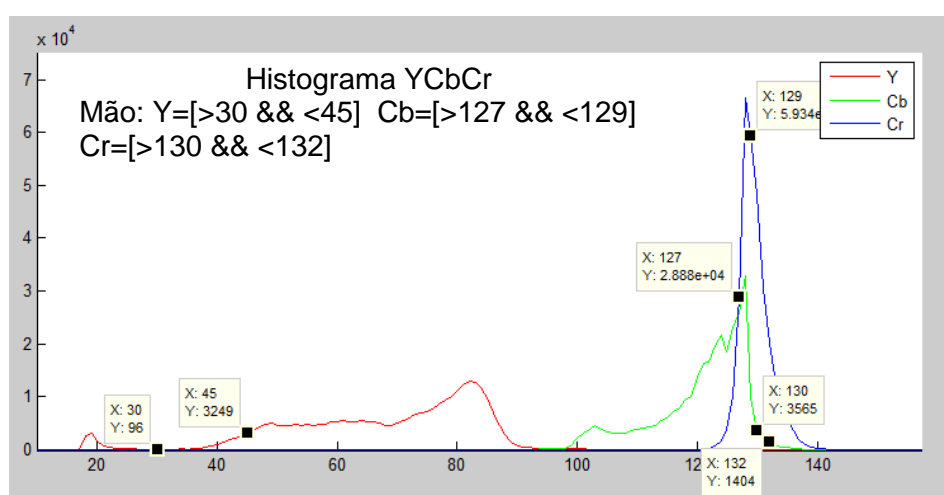


Figura 142 – Histograma YCbCr da mão número 6.

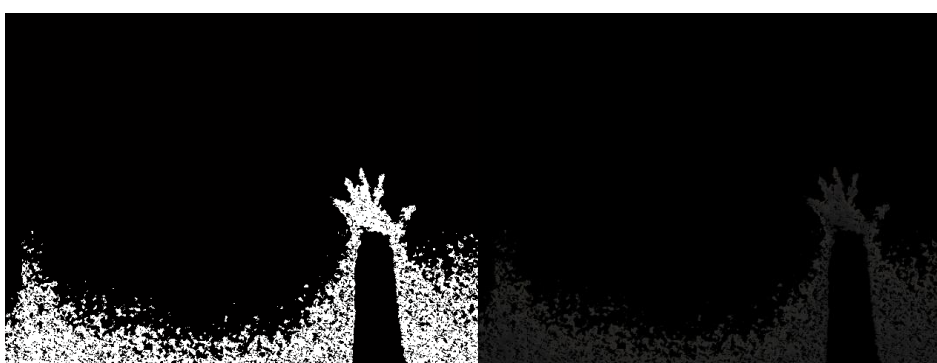


Figura 143 – Segmentação YCbCr da mão número 6.

A.1.6.3 Método Probabilístico

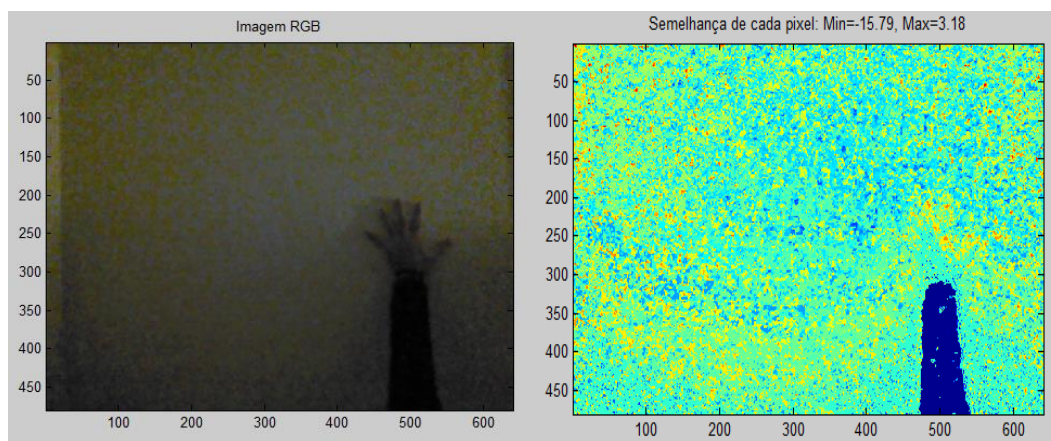


Figura 144 – Transformação de RGB para método probabilístico da mão número 6.

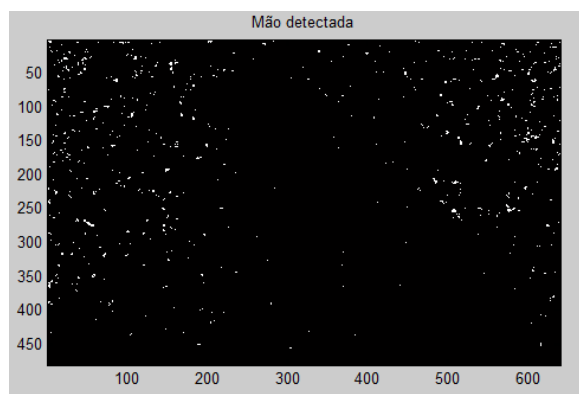


Figura 145 – Resultado das operações morfológicas na imagem binarizada da mão número 6.